Qualcomm

Hard and Soft equivariance priors via Steerable CNNs

Gabriele Cesa

Qualcomm AI Research*

University of Amsterdam, AMLab

04/09/2024

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Hard and Soft equivariance priors via Steerable CNNs

First, due credit to my amazing collaborators!

Maurice Weiler* and Gabriele Cesa*. General E(2)-Equivariant Steerable CNNs, *Neural Information Processing Systems, 2019*

Gabriele Cesa, Leon Lang, Maurice Weiler, A Program to build E(n)-Equivariant Steerable CNNs, *International Conference on Representation Learning, 2022*

Maksim Zhdanov, Nico Hoffmann, Gabriele Cesa, Implicit Convolutional Kernels for Steerable CNNs*. Neural Information Processing Systems, 2023*

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Agenda

From Group Convolution to Steerable Filters

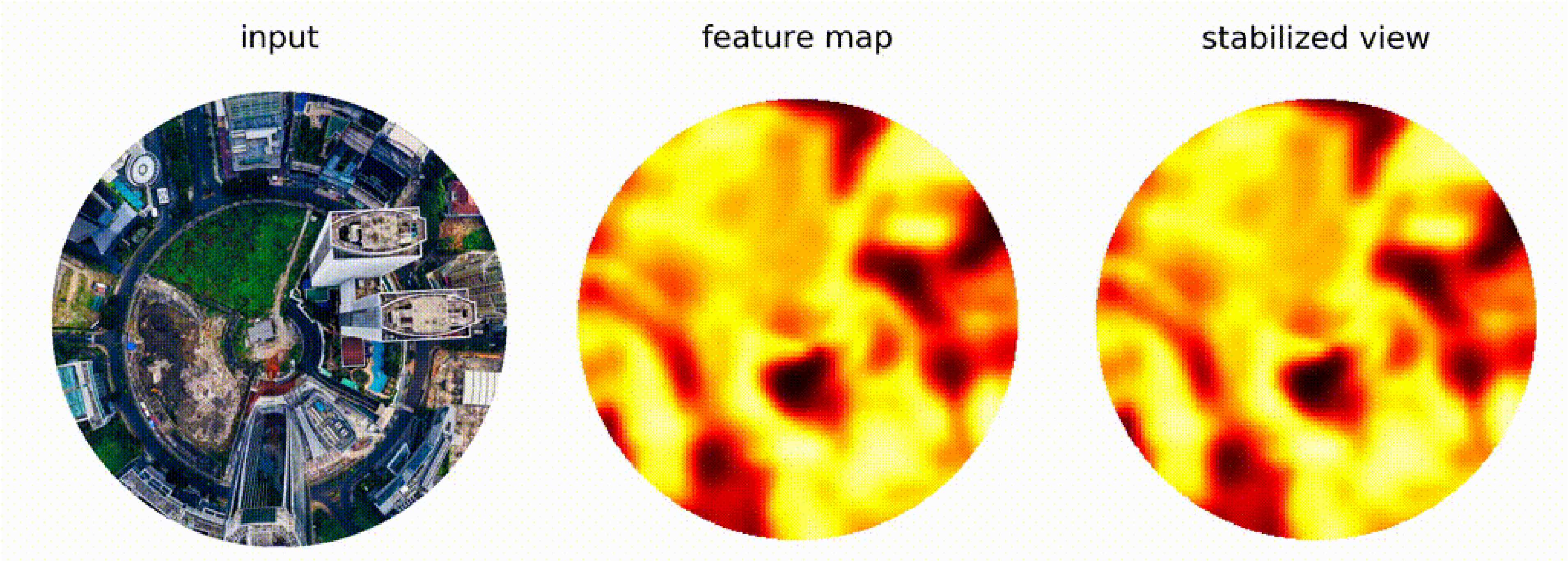
Steerable Fields and Representation Theory

Steerable CNNs

Hard Priors: solving the exact kernel constraint

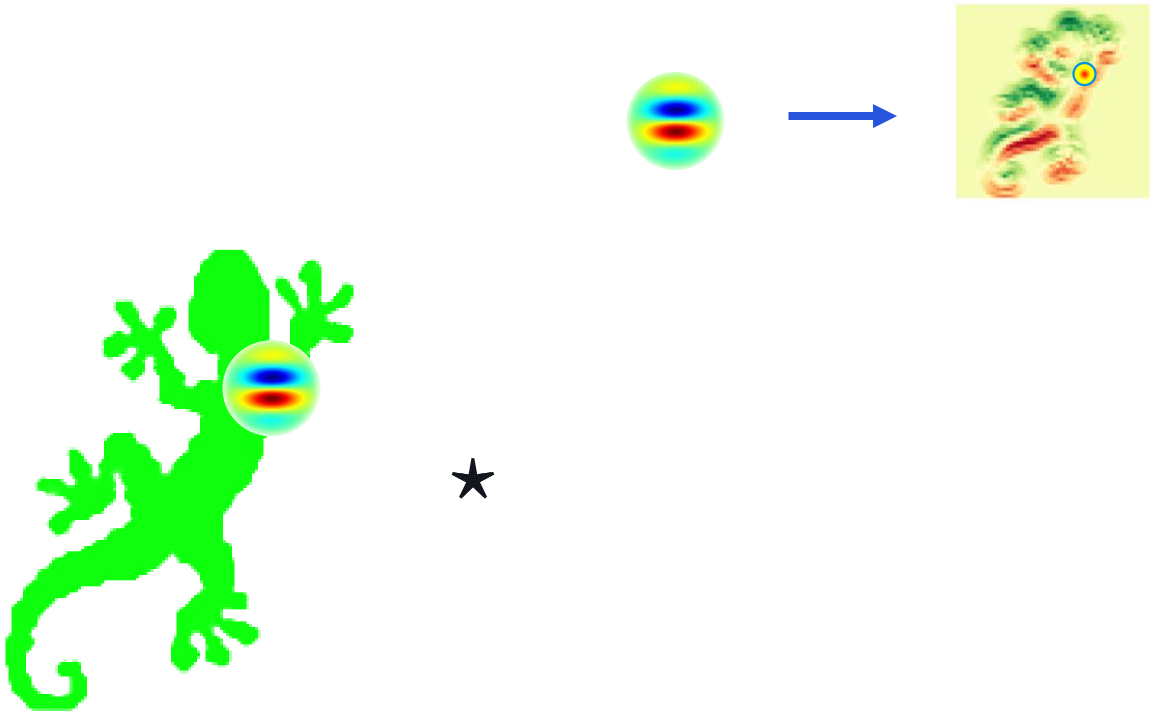
Soft Priors: learnable kernel constraint

Equivariance: CNN (rotation equivariance?)

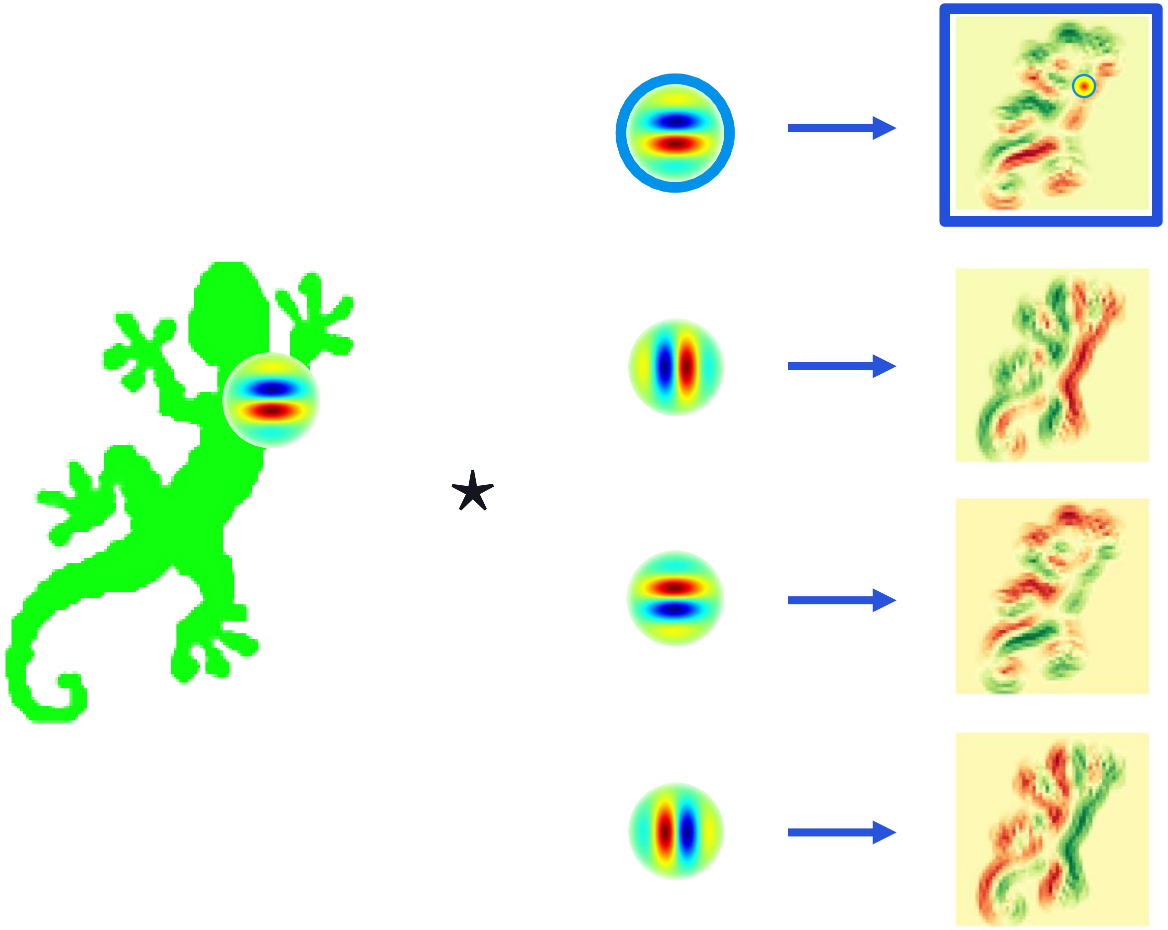


Source: e2cnn

Convolution: $(\mathbb{R}^n, +)$ - equivariance

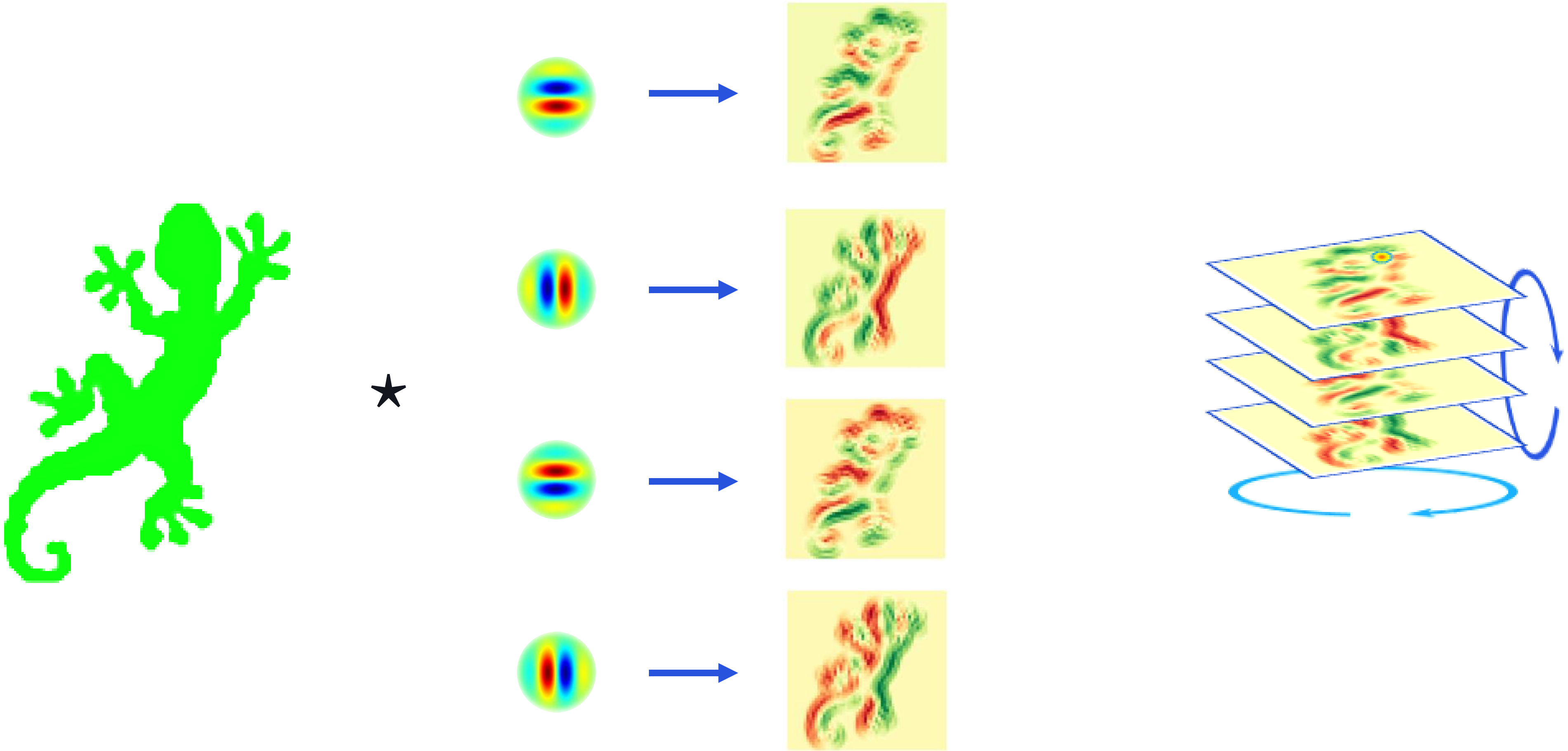


Group Convolution: $(\mathbb{R}^n, +) \rtimes C_4$ - equivariance



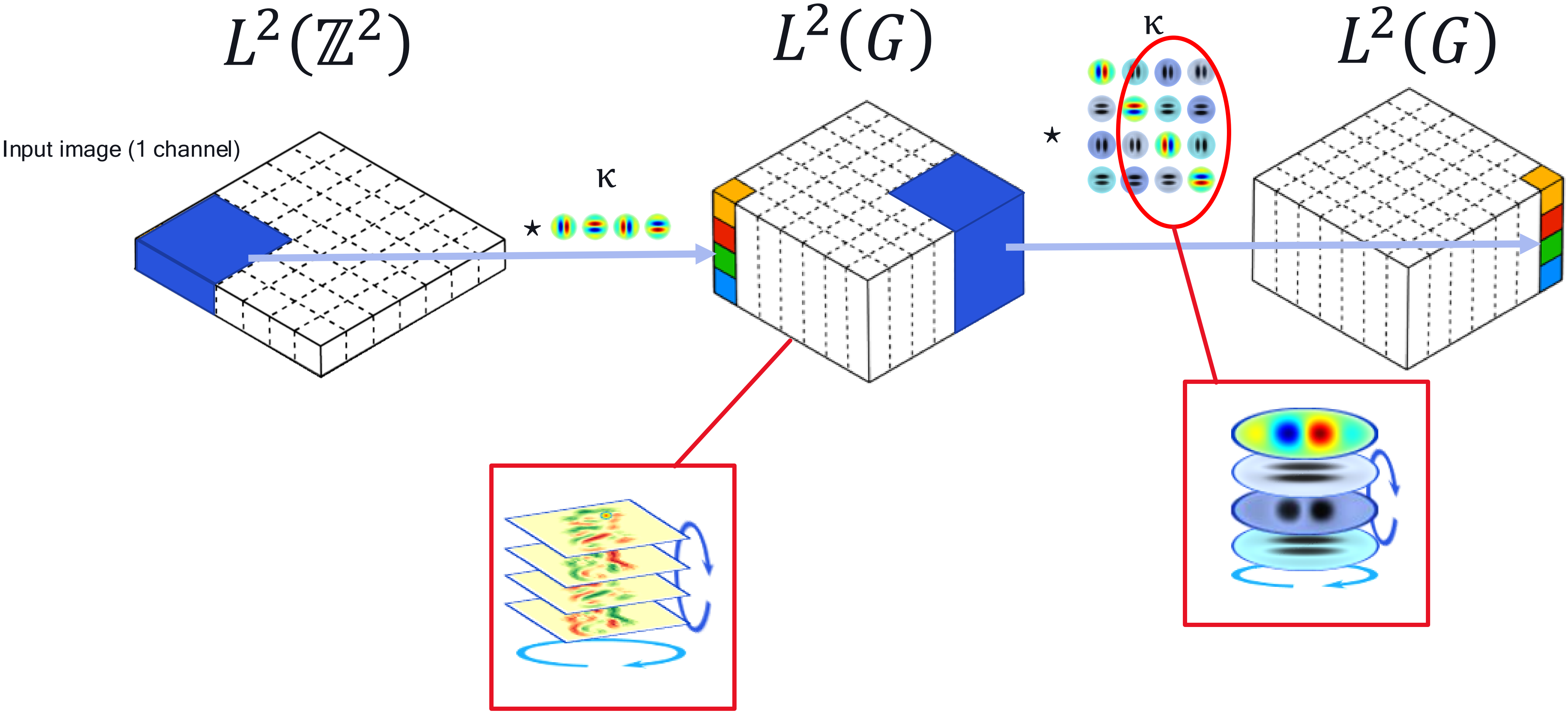
Taco S. Cohen and Max Welling, Group Equivariant Convolutional Networks,
International Conference on Machine Learning (ICML), 2016

Group Convolution (**lifting convolution**): $(\mathbb{R}^n, +) \rtimes C_4$ - equivariance



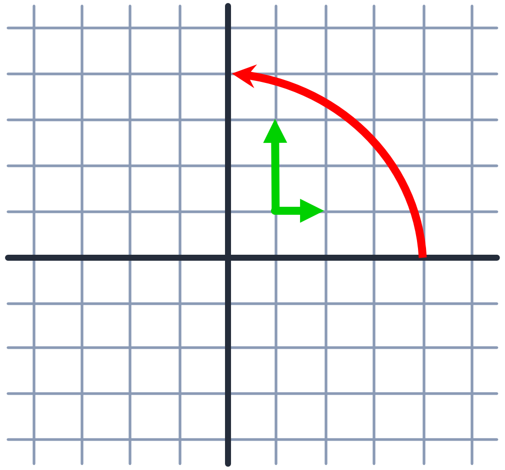
Taco S. Cohen and Max Welling, Group Equivariant Convolutional Networks,
International Conference on Machine Learning (ICML), 2016

Group Convolution



Taco S. Cohen and Max Welling, Group Equivariant Convolutional Networks,
International Conference on Machine Learning (ICML), 2016

Group Convolution: Challenges



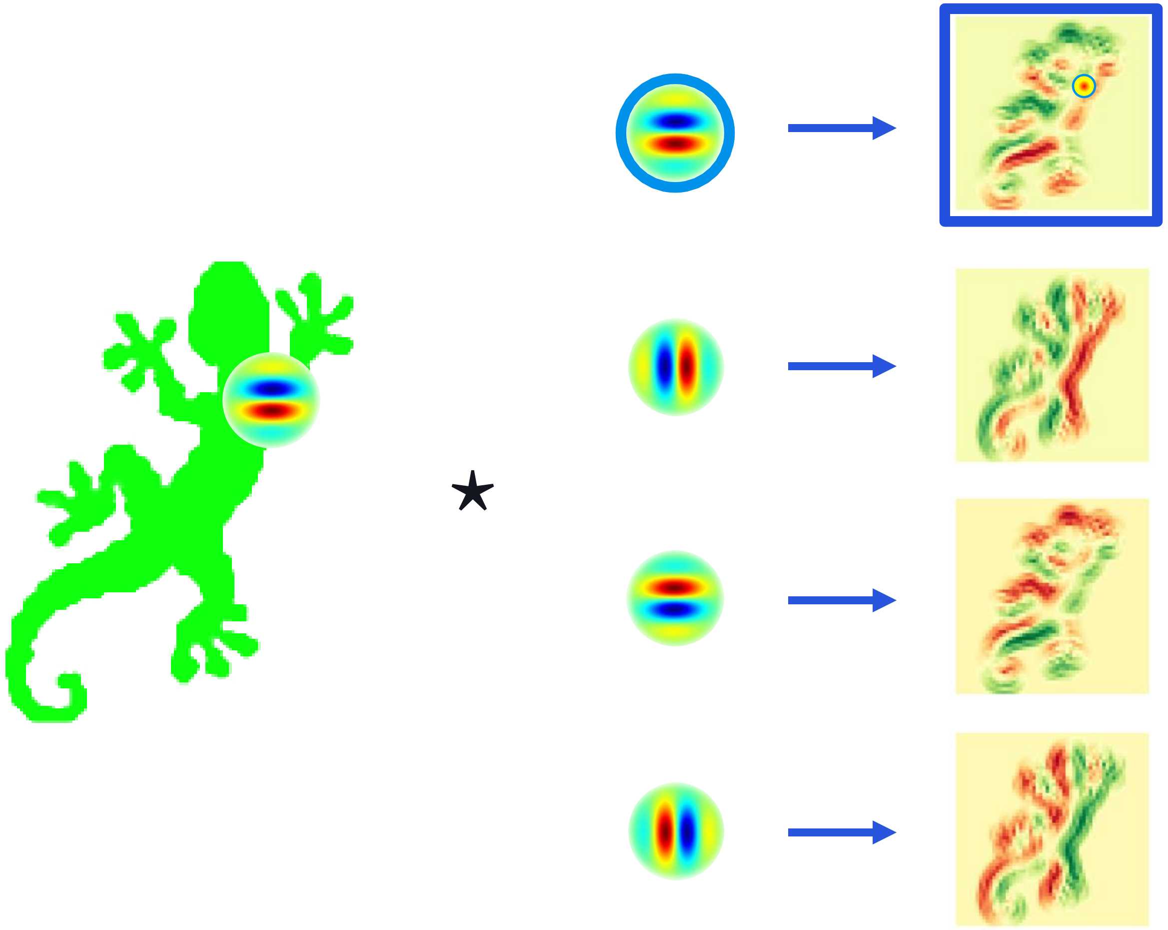
$G = (\mathbb{Z}^2, +) \rtimes C_4$

$$[\kappa \star f](g) = \int_{x \in X} \kappa(g^{-1}\boldsymbol{x})\ f(\boldsymbol{x}) d\mu(x)$$

- Often some form of discretization is required (potential artifacts!)
- What if input or output are not in $L^2(\mathbb{R}^n)$ or $L^2(G)$?
  - e.g. vector fields?
- The group G might be infinite or too large to fully enumerate
  - $G = SO(3)$ rotations of a point cloud
  - $n!$ permutations of a set or nodes of a graph

Group Convolution

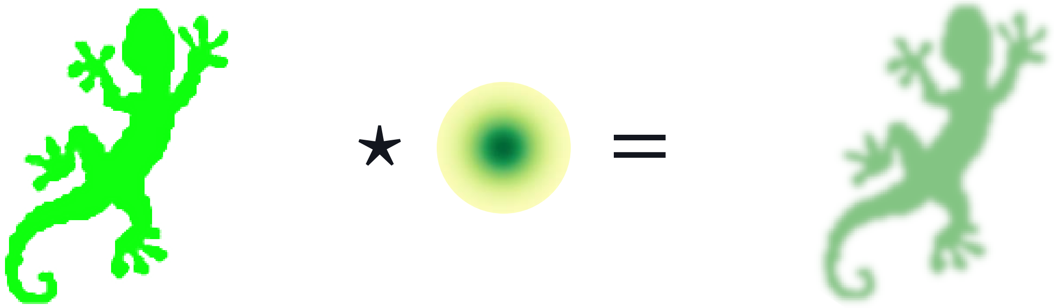
$$[\kappa \star \cdot]: L^2(\mathbb{Z}^2) \rightarrow L^2((\mathbb{Z}^2, +) \rtimes C_4)$$



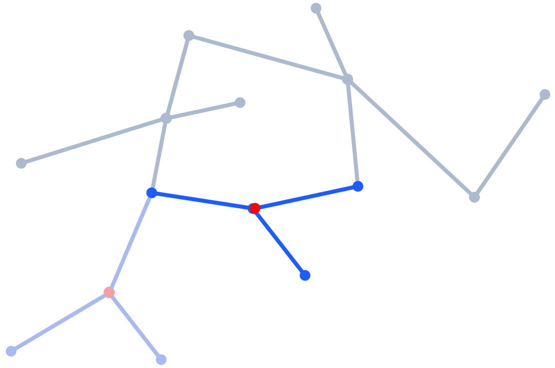
Isotropic Filters

$$[\kappa \star \cdot]: L^2(\mathbb{Z}^2) \to L^2(\mathbb{Z}^2)$$

- Isotropic filters: the response does not change when rotated



- Not very expressive
- Analogous to Graph Message Passing: no directional dependence!



Steerable Filters

$$
\begin{aligned}
[R_\theta . \kappa](r,\phi) &= \kappa(r, \phi - \theta) \\
&= w(r)\cos(\phi + \beta - \theta) \\
&= w(r)\cos(\phi + \beta)\cos\theta - w(r)\sin(\phi + \beta)\sin\theta \\
&= \cos\theta\,\kappa(r,\phi) - \sin\theta\,\kappa\left(r, \phi - \frac{\pi}{2}\right) \\
&= \cos\theta\,\kappa(r,\phi) - \sin\theta\,R_{\frac{\pi}{2}} . \kappa(r,\phi)
\end{aligned}
$$

- Filter can be rotated via linear combination

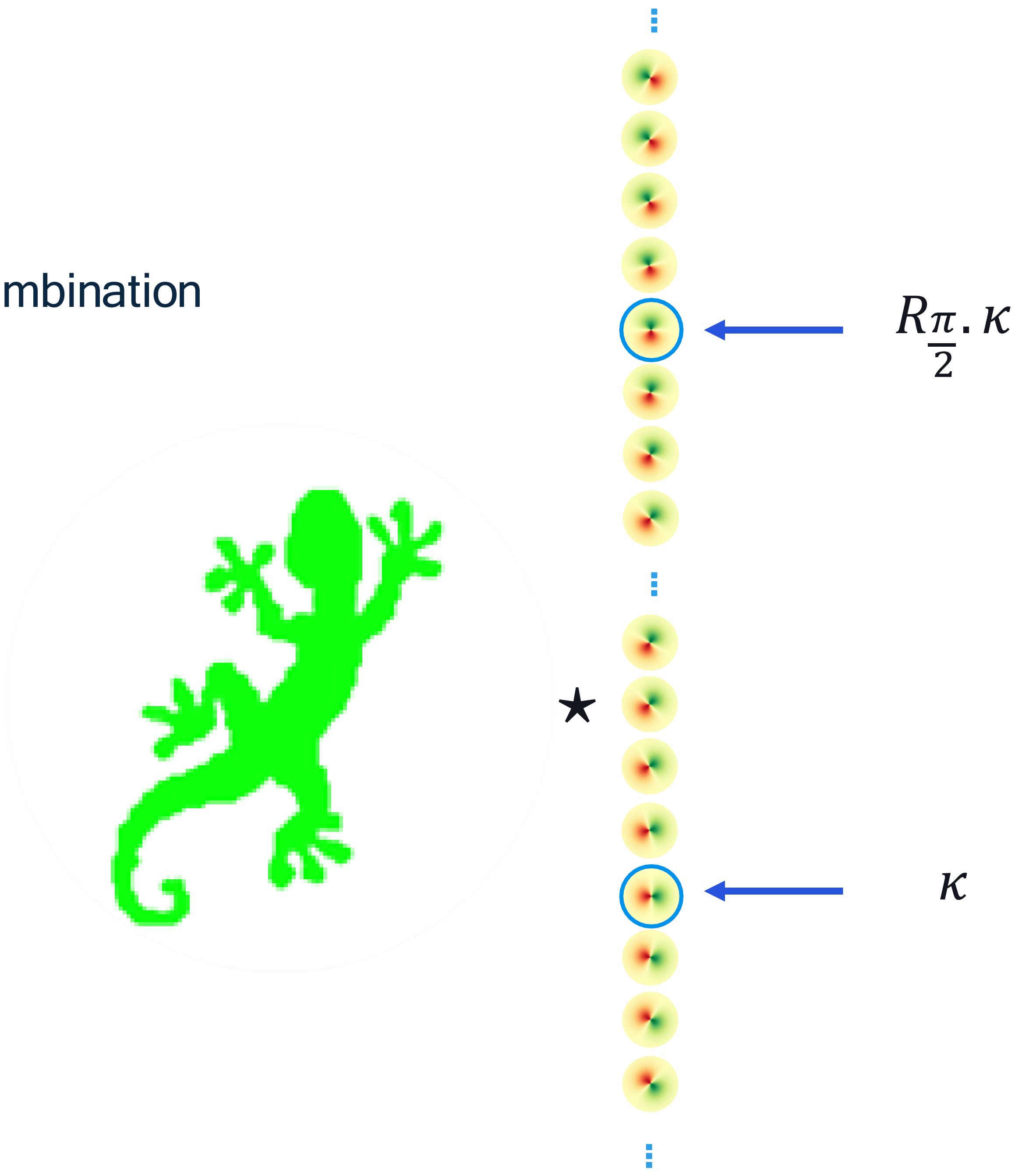
$$\kappa(r,\phi) = w(r)\cos(\phi + \beta)$$

- e.g. filters used for edge-detection in classical Computer Vision

William T. Freeman and Edward H. Adelson. The design and use of steerable filters.
IEEE Transactions on Pattern Analysis & Machine Intelligence, 1991

Steerable Filters

- Filter can be rotated via linear combination



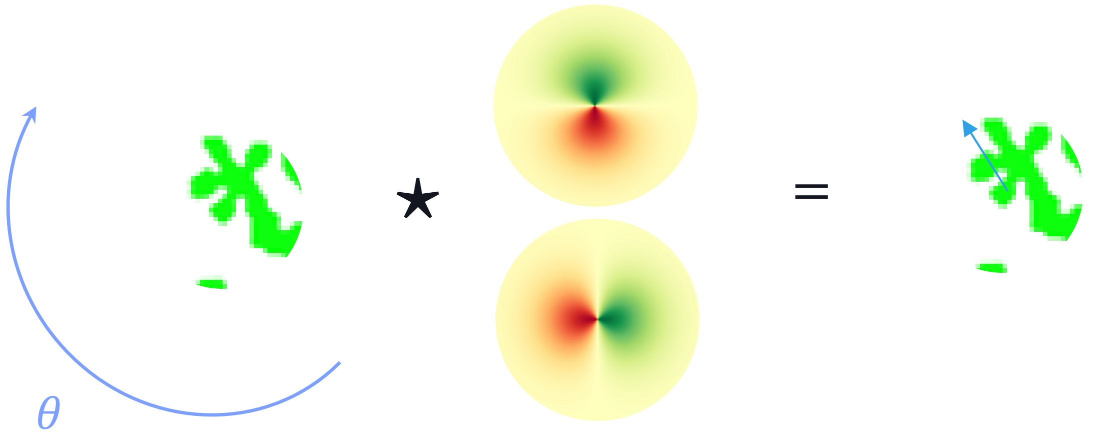
Thanks to *linearity* of convolution operator:

$$R_\theta.\kappa \star f = \left(\cos\theta\,\kappa - \sin\theta\, R_{\frac{\pi}{2}}.\kappa\right) \star f$$
$$= \cos\theta\,(\kappa \star f) - \sin\theta\,\left(R_{\frac{\pi}{2}}.\kappa \star f\right)$$

Steerable Filters

- Filter can be rotated via linear combination

$$\rho(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

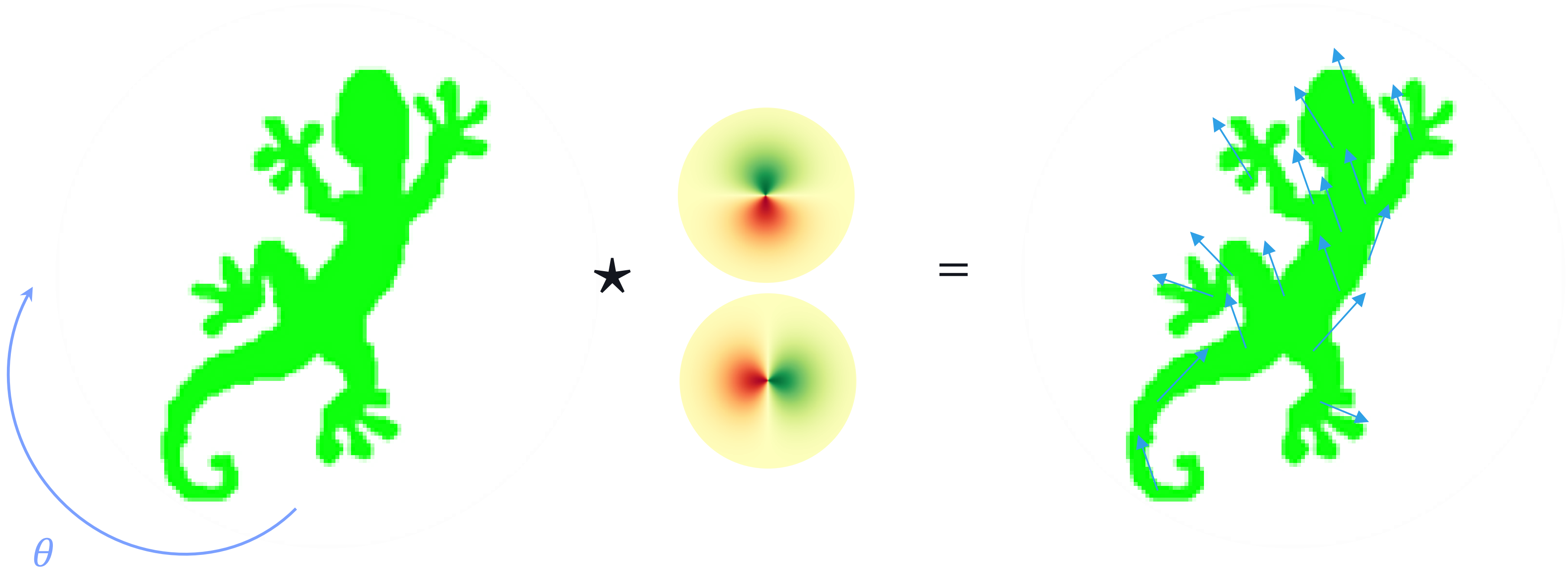


$$R_\theta . \kappa \star f = \cos\theta\,(\kappa \star f) - \sin\theta \left(R_{\frac{\pi}{2}} . \kappa \star f\right)$$

Steerable Filters

- Filter can be rotated via linear combination

$$\rho(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

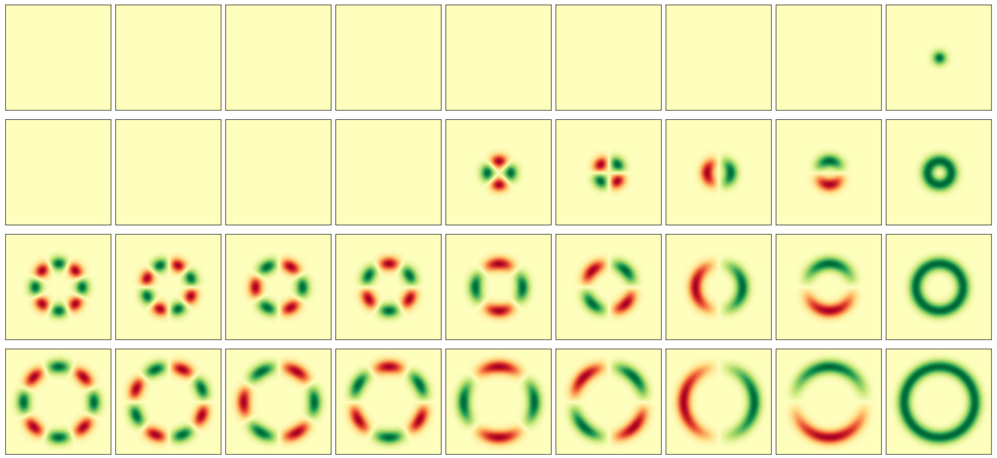


$$R_\theta . \kappa \star f = \cos\theta\,(\kappa \star f) - \sin\theta\left(R_{\frac{\pi}{2}} . \kappa \star f\right)$$

Steerable Filters: Other examples

- Filter can be rotated via linear combination

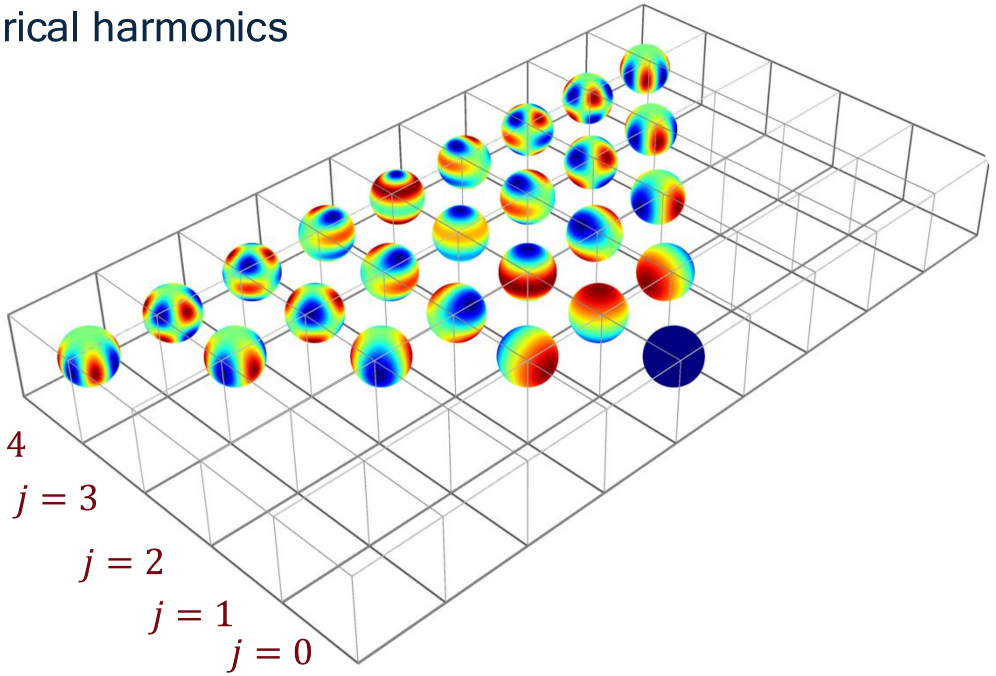
Circular harmonics



$j = 4$ $\quad j = 3$ $\quad j = 2$ $\quad j = 1$ $\quad j = 0$

$$\rho_j(\theta) = \begin{bmatrix} \cos j\theta & -\sin j\theta \\ \sin j\theta & \cos j\theta \end{bmatrix}$$

Spherical harmonics



$j = 4$
$j = 3$
$j = 2$
$j = 1$
$j = 0$

$D^j(\alpha, \beta, \gamma) \in \mathbb{R}^{2j+1 \times 2j+1}$ Wigner D-matrix

Agenda

From Group Convolution to Steerable Filters

Steerable Fields and Representation Theory

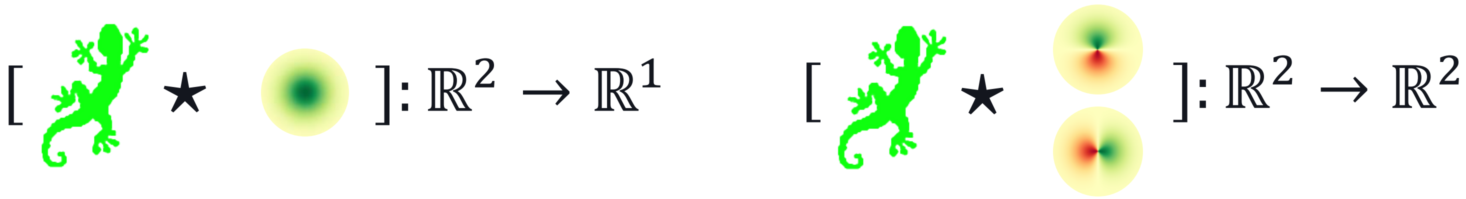
Steerable CNNs

Hard Priors: solving the exact kernel constraint

Soft Priors: learnable kernel constraint

Feature Fields

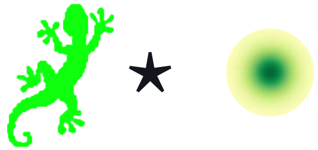
- Interpret features as a multi-channels signal $f: \mathbb{R}^n \to \mathbb{R}^d$
- Signal transforms under *(point) symmetry group* G according to a *transformation law*
  - symmetry group G : e.g. rotations or reflections
  - N.B.: before we used G to also indicate translations
  - For now, we will implicitly consider equivariance to $(\mathbb{R}^n, +) \rtimes G$

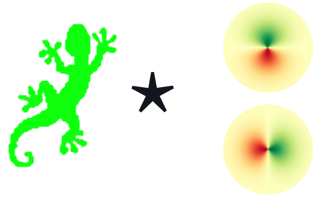


Feature Fields and Steerable CNNs

- Interpret features as a multi-channels signal $f: \mathbb{R}^n \to \mathbb{R}^d$
- Signal transforms under *(point) symmetry group* G according to a *transformation law*
  - symmetry group G : e.g. rotations or reflections
- Type of transformation identified by a **representation** of G $\rho: G \to \mathbb{R}^{d \times d}$

$$[g.f](x) = \rho(g) f(g^{-1}x)$$

[]: $\mathbb{R}^2 \to \mathbb{R}^1$

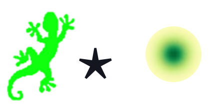
[]: $\mathbb{R}^2 \to \mathbb{R}^2$

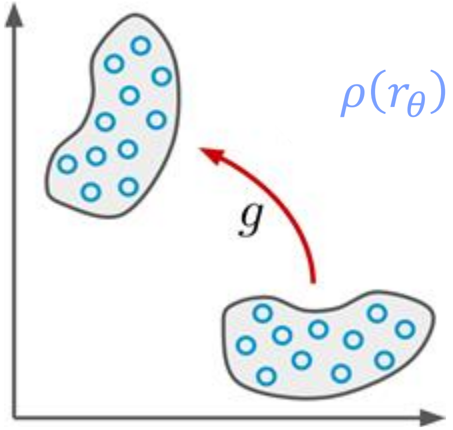
Taco S. Cohen and Max Welling, Steerable CNNs,
International Conference on Learning Representations (ICLR), 2017

Feature Fields and Steerable CNNs

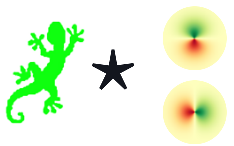
- Interpret features as a multi-channels signal $f: \mathbb{R}^n \to \mathbb{R}^d$
- Signal transforms under *(point) symmetry group* G according to a *transformation law*
  - symmetry group G : e.g. rotations or reflections
- Type of transformation identified by a **representation** of G $\quad \rho: G \to \mathbb{R}^{d \times d}$

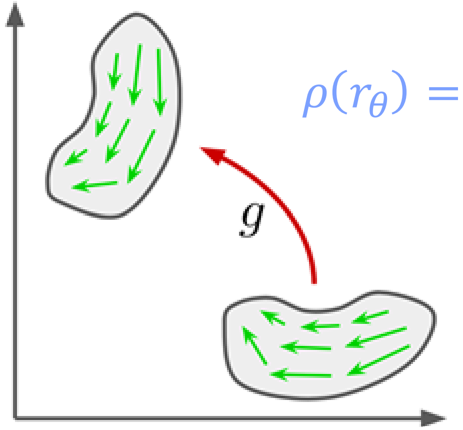
$$[g.f](x) = \rho(g) f(g^{-1}x)$$

$f = [$  $]: \mathbb{R}^2 \to \mathbb{R}^1$



$\rho(r_\theta) = 1$

$f = [$  $]: \mathbb{R}^2 \to \mathbb{R}^2$



$\rho(r_\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Source: e2cnn

Definition: **Representation** of a Compact Group

A *(real) representation* $\rho: G \to GL(\mathbb{R}^d)$ of a *compact group* G is a map which associates to each element $g \in G$ an *invertible* $d \times d$ matrix s.t.:

- $\forall a, b \quad \rho(a)\rho(b) = \rho(ab)$
- $\forall a \ \ \rho(a)^{-1} = \rho(a^{-1})$
- $\rho(e) = \mathrm{Id}_{d\times d}$

Definition: **Representation** of a Compact Group

A *(real) representation* $\rho: G \to GL(\mathbb{R}^d)$ of a *compact group* G is a map which associates to each element $g \in G$ an *invertible* $d \times d$ matrix s.t.:

- $\forall a, b \quad \rho(a)\rho(b) = \rho(ab)$
- $\forall a \ \ \rho(a)^{-1} = \rho(a^{-1})$
- $\rho(e) = \mathrm{Id}_{d\times d}$

Can assume w.l.o.g. *orthogonal* representations, i.e. that $\rho(g)^{-1} = \rho(g)^T$

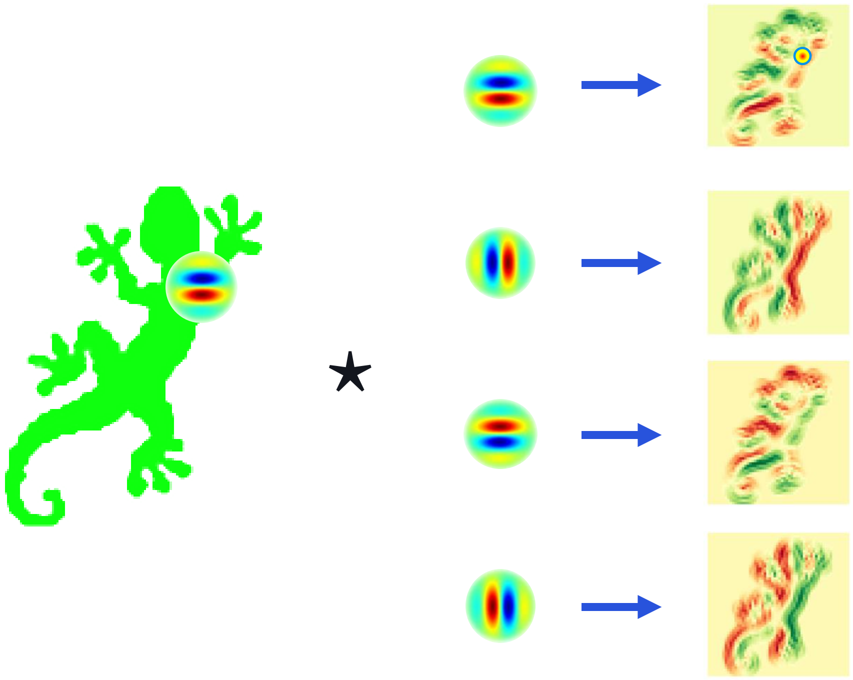
Describes the action of a group G on a *vector space* $V = \mathbb{R}^d$

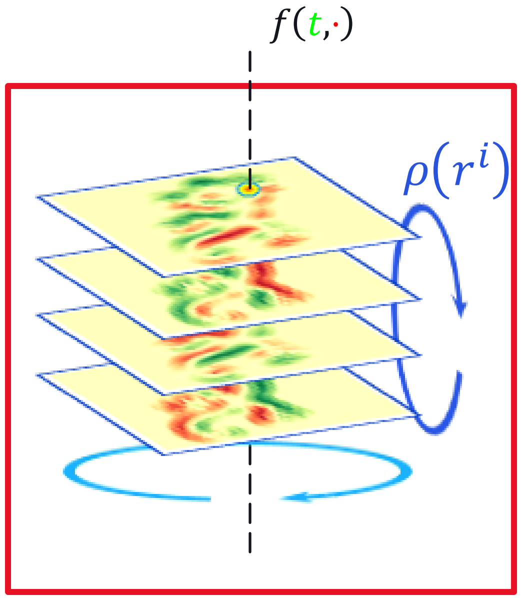
E.g. representations of the rotation group $SO(2)$

- *Trivial representation* $\rho(r_\theta) = 1$
- *Standard representation* $\rho(r_\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Special Case: Group Convolution

$$f(t,\cdot): C_4 \to \mathbb{R}$$





$$\rho(r^0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

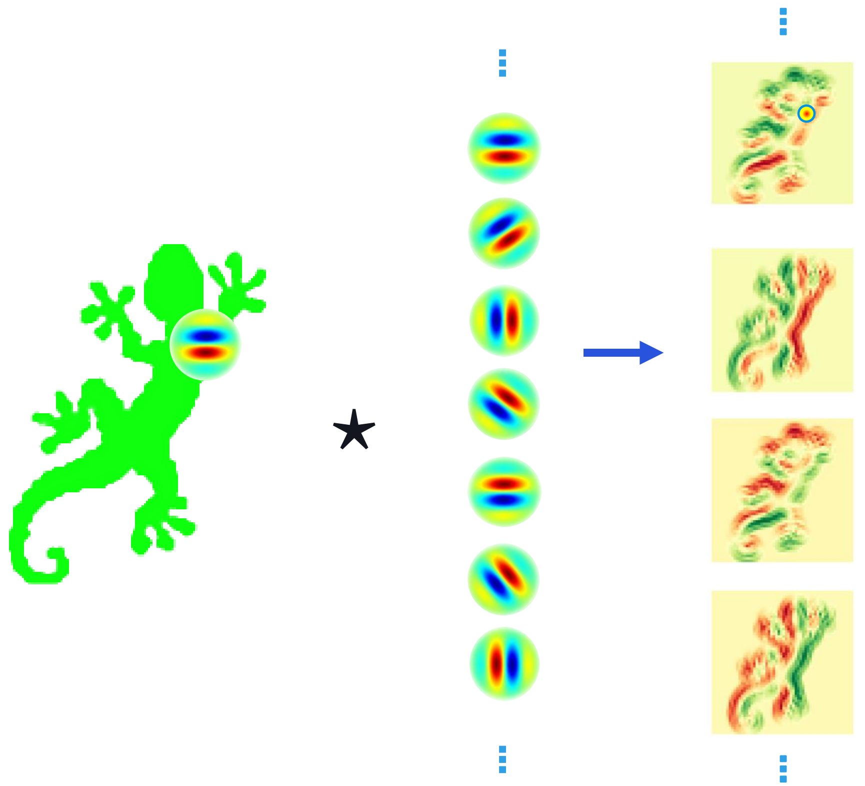
$$\rho(r^1) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\rho(r^2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

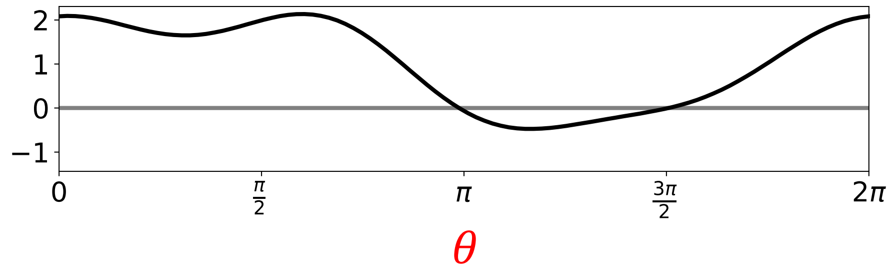
$$\rho(r^3) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Special Case: Group Convolution

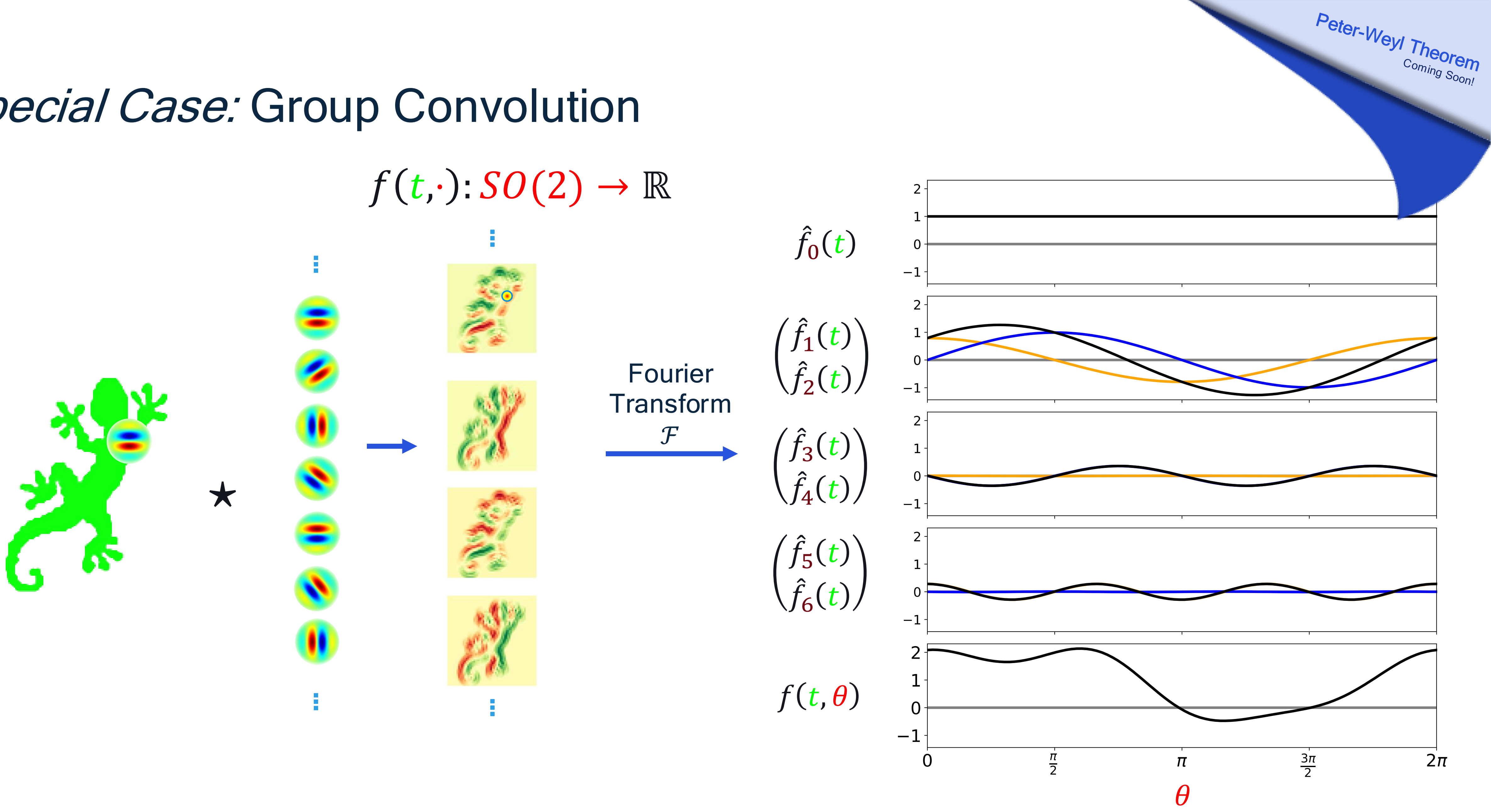
$f(t, \cdot): SO(2) \to \mathbb{R}$



$f(t, \theta)$

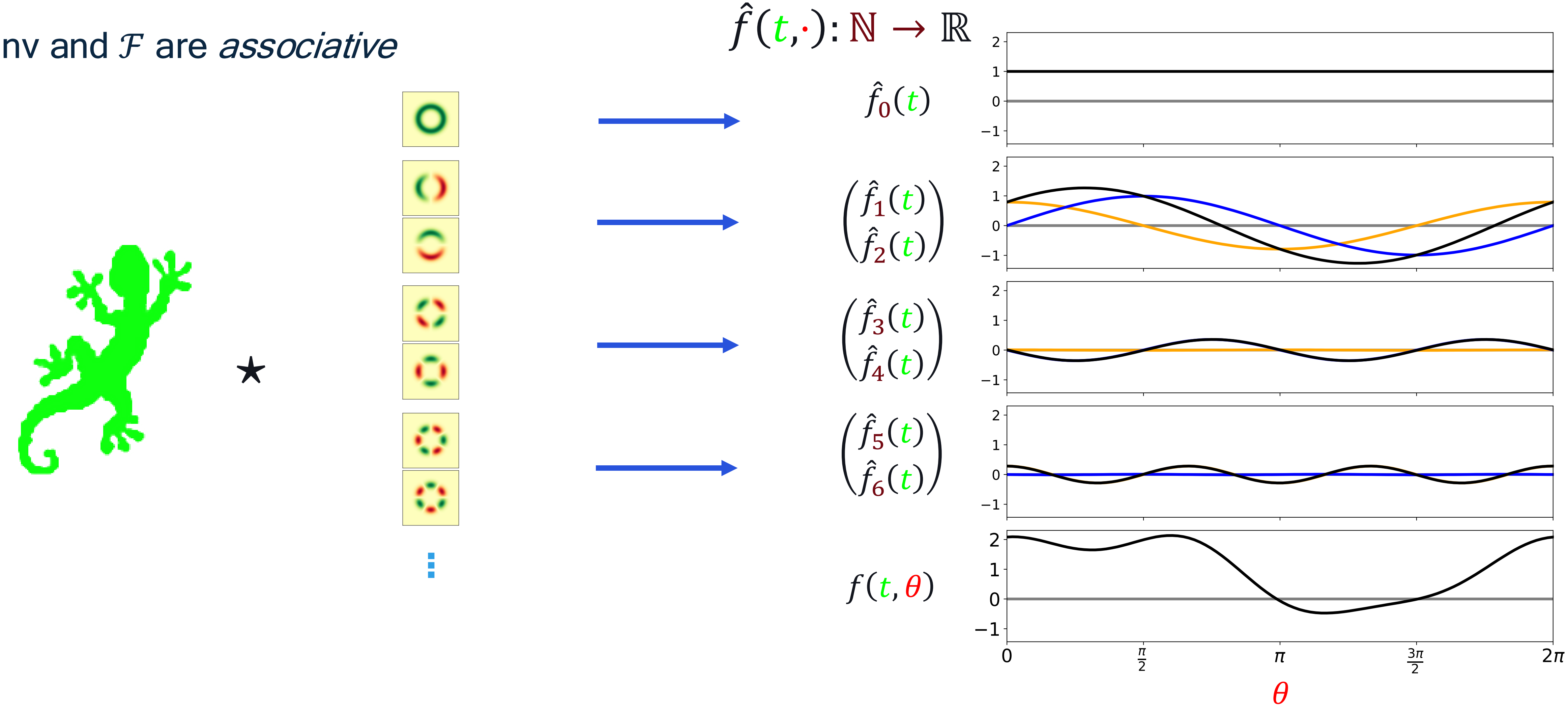


Special Case: Group Convolution



Special Case: Group Convolution

Conv and $\mathcal{F}$ are *associative*



Direct Sum

$$\rho(\theta) = \begin{bmatrix} 1 & & & \\ & \begin{bmatrix} \cos 1\theta & -\sin 1\theta \\ \sin 1\theta & \cos 1\theta \end{bmatrix} & & \\ & & \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix} & \\ & & & \begin{bmatrix} \cos 3\theta & -\sin 3\theta \\ \sin 3\theta & \cos 3\theta \end{bmatrix} \end{bmatrix}$$

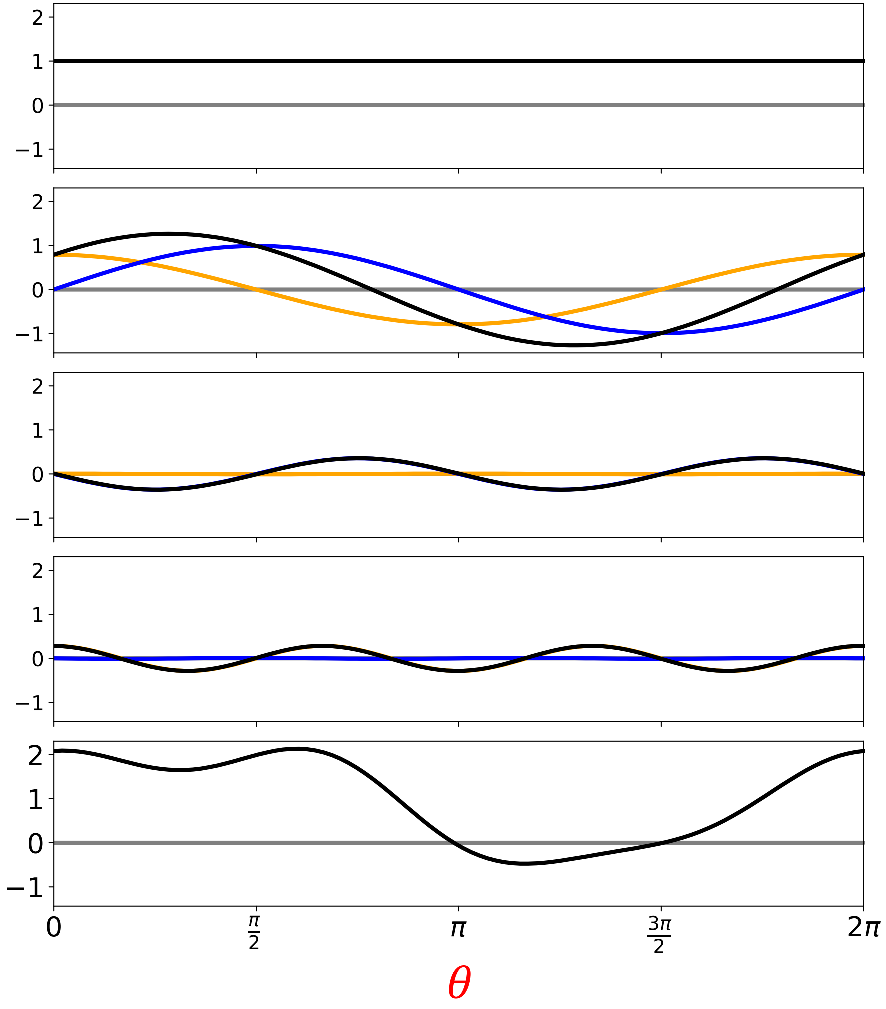
$\hat{f}_0(t)$

$\begin{pmatrix} \hat{f}_1(t) \\ \hat{f}_2(t) \end{pmatrix}$

$\begin{pmatrix} \hat{f}_3(t) \\ \hat{f}_4(t) \end{pmatrix}$

$\begin{pmatrix} \hat{f}_5(t) \\ \hat{f}_6(t) \end{pmatrix}$

$f(t, \theta)$



$$\rho(\theta) = \bigoplus_k \rho_k(\theta)$$

$$\bigoplus_k \begin{pmatrix} \hat{f}_{2k}(t) \\ \hat{f}_{2k+1}(t) \end{pmatrix}$$

Formally: the Regular Representation

- $L^2(G)$ is the vector space of *square integrable functions* on G

- $L^2(G)$ carries an **orthogonal action** of G

$$g: L^2(G) \to L^2(G), \qquad f \mapsto g.f$$
$$[g.f](\boldsymbol{x}) := f(g^{-1}\,\boldsymbol{x})$$

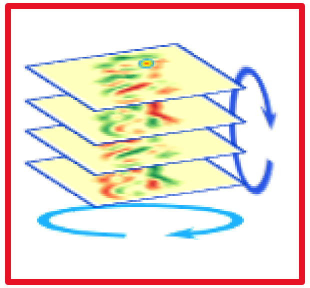
- This is the *Regular Representation* of G

Formally: the Regular Representation

- $L^2(G)$ is the vector space of *square integrable functions* on G
- $L^2(G)$ carries an **orthogonal action** of G

$$g: L^2(G) \to L^2(G), \qquad f \mapsto g.f$$
$$[g.f](\boldsymbol{x}) := f(g^{-1}\,\boldsymbol{x})$$

- This is the *Regular Representation* of G
- When G is a **finite group** it looks like permutation matrices (e.g. C_4)



$$\rho(r^0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \rho(r^1) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \rho(r^2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \rho(r^3) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

One Theorem To Rule Them All: Peter-Weyl theorem

- Let G be a *compact* group
- There is a set of *irreducible representations (irreps)* denoted $\hat{G}$
  - analogous to frequencies in classical Fourier Transform
  - e.g. circular harmonics or Wigner-D Matrices

The matrix coefficients of the *irreps* form an *orthogonal basis* for $L^2(G)$

$$f(g) = \sum_{\psi \in \hat{G}} \sqrt{d_\psi} \sum_{1 \leq ij \leq d_\psi} \hat{f}(\psi)_{ij} \psi(g)_{ij}$$

Fourier Transform

$$\psi_0(\theta) = 1 \quad \psi_1(\theta) = \begin{bmatrix} \cos 1\theta & -\sin 1\theta \\ \sin 1\theta & \cos 1\theta \end{bmatrix} \quad \psi_2(\theta) = \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix} \quad \psi_3(\theta) = \begin{bmatrix} \cos 3\theta & -\sin 3\theta \\ \sin 3\theta & \cos 3\theta \end{bmatrix}$$

Peter-Weyl Theorem

One Theorem To Rule Them All: Peter-Weyl theorem

- Let G be a *compact* group
- There is a set of *irreducible representations (irreps)* denoted $\hat{G}$
  - analogous to frequencies in classical Fourier Transform
  - e.g. circular harmonics or Wigner-D Matrices

Fourier Transform

The matrix coefficients of the *irreps* form an *orthogonal basis* for $L^2(G)$

$$f(g) = \sum_{\psi \in \hat{G}} \sqrt{d_\psi} \sum_{1 \leq ij \leq d_\psi} \hat{f}(\psi)_{ij} \psi(g)_{ij}$$

$\hat{f}(\psi) \in \mathbb{R}^{d_\psi \times d_\psi}$
Contains the weights

One Theorem To Rule Them All: Peter-Weyl theorem

- Let G be a *compact* group
- There is a set of *irreducible representations (irreps)* denoted $\hat{G}$
  - analogous to frequencies in classical Fourier Transform
  - e.g. circular harmonics or Wigner-D Matrices

Fourier Transform

The matrix coefficients of the *irreps* form an *orthogonal basis* for $L^2(G)$

$$f(g) = \sum_{\psi \in \hat{G}} \sqrt{d_\psi} \sum_{1 \leq ij \leq d_\psi} \hat{f}(\psi)_{ij} \psi(g)_{ij}$$

Caveat: this is actually a basis only in $\mathbb{C}$, but sometimes it has redundant entries in $\mathbb{R}$, e.g.

$$\psi_1(\theta) = \begin{bmatrix} \cos 1\theta & -\sin 1\theta \\ \sin 1\theta & \cos 1\theta \end{bmatrix}$$

One Theorem To Rule Them All: Peter-Weyl theorem (Pt. 2)

- Let G be a *compact* group
- There is a set of *irreducible representations (irreps)* denoted $\hat{G}$
  - analogous to frequencies in classical Fourier Transform
  - e.g. circular harmonics or Wigner-D Matrices

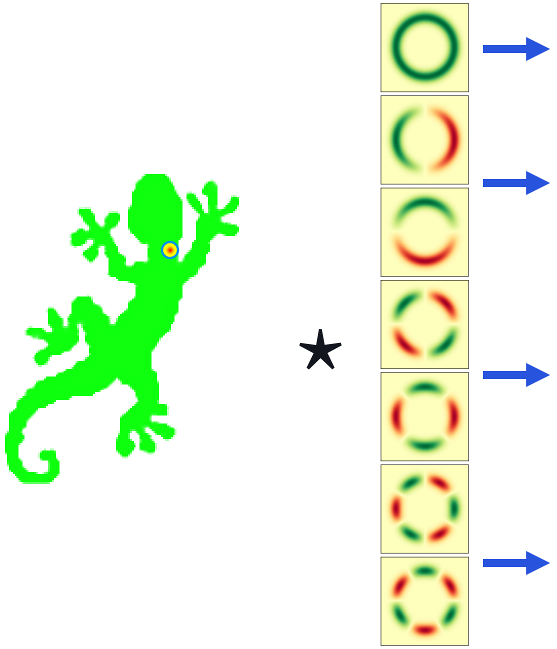
Any unitary representation ρ can be decomposed as a *direct sum* of *irreps* up to a change of basis Q_ρ

$$\rho(g) = Q_\rho^T \left(\bigoplus_{\psi_i \in \hat{G}} \bigoplus_{r}^{[i(\rho)]} \psi_i(g) \right) Q_\rho$$

$$\rho(\theta) = \begin{bmatrix} 1 & & \\ & \begin{bmatrix} \cos 1\theta & -\sin 1\theta \\ \sin 1\theta & \cos 1\theta \end{bmatrix} & \\ & & \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix} \end{bmatrix}$$

What did we find?

- Group convolution with steerable filters produces smaller steerable features
  - No need to store redundant activations
- So far, only studied lifting convolution $L^2(\mathbb{R}^n) \to L^2\big((\mathbb{R}^n, +) \rtimes G\big)$
  - Input is a **scalar field**
  - Recover lifting convolution when using output regular representation
- What about other input fields? Intermediate layers?



$$\hat{f}_0(t) \quad \begin{pmatrix} \hat{f}_1(t) \\ \hat{f}_2(t) \end{pmatrix} \quad \begin{pmatrix} \hat{f}_3(t) \\ \hat{f}_4(t) \end{pmatrix} \quad \begin{pmatrix} \hat{f}_5(t) \\ \hat{f}_6(t) \end{pmatrix}$$

$$[g.f](x) = \rho(g) f(g^{-1}x)$$

Agenda

From Group Convolution to Steerable Filters

Steerable Fields and Representation Theory

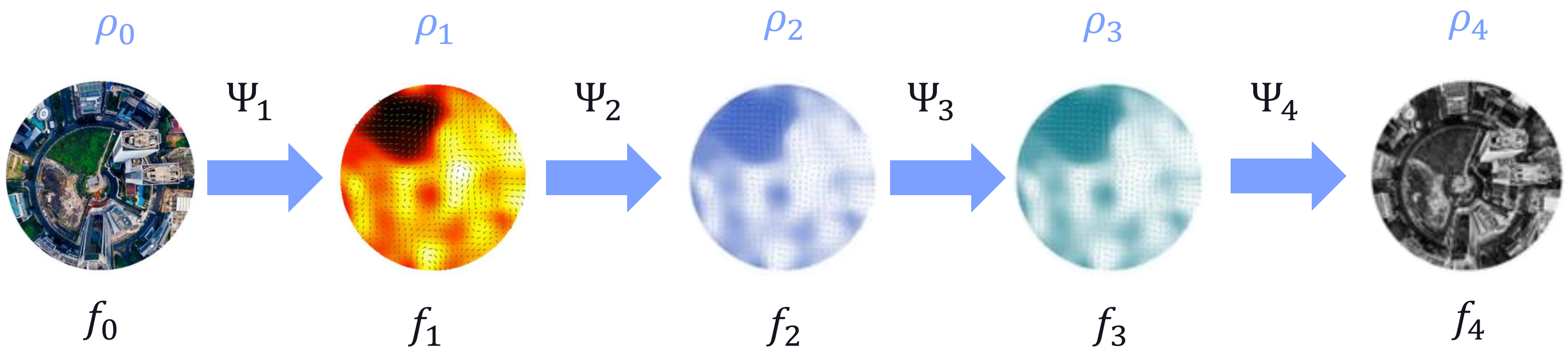
Steerable CNNs

Hard Priors: solving the exact kernel constraint

Soft Priors: learnable kernel constraint

Feature Fields and Steerable CNNs

- Symmetry group G
- An intermediate feature is a multi-channels signal $f_l: \mathbb{R}^n \to \mathbb{R}^d$
- Associated with its own transformation law ρ_l
- Steerable CNN is equivariant when each layer Ψ_l commutes with its input and output transformations



Problem

Given a choice of input and output **steerable feature types**, what convolution do we need to use?

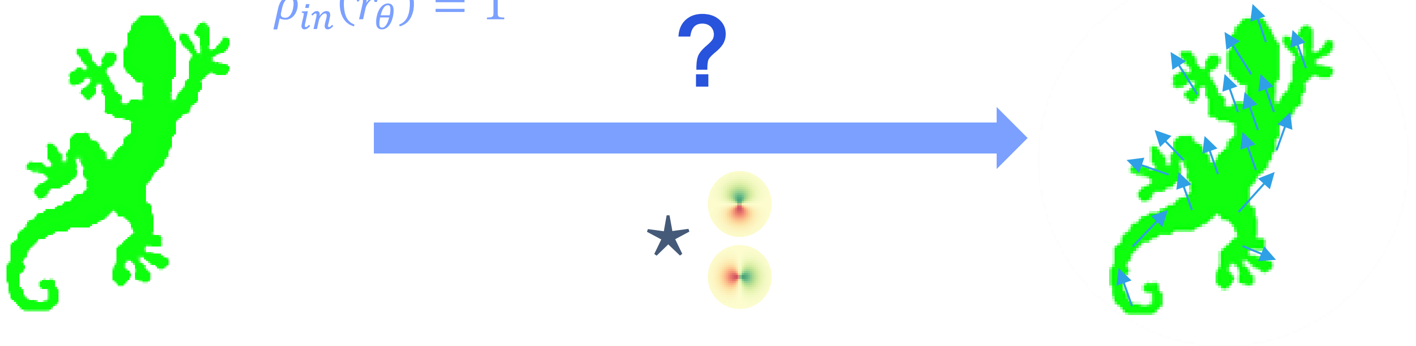
- What kind of filters produce an output feature map with the *desired transformation type*?

$f_{in}: \mathbb{R}^2 \to \mathbb{R}^1$

$f_{out}: \mathbb{R}^2 \to \mathbb{R}^2$

$\rho_{in}(r_\theta) = 1$

$\rho_{out}(r_\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$



Steerable CNNs

- Standard ***convolution*** with G *-steerable filter* K guarantees also G **equivariance**

$$K: \mathbb{R}^n \rightarrow \mathbb{R}^{d_{out} \times d_{in}}$$

$$K(x) = \rho_{out}(g) K(g^{-1}.x) \rho_{in}(g)^T$$

Steerability Constraint

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma and Taco Cohen,
3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data,
Conference on Neural Information Processing Systems (NIPS), 2018

Taco Cohen, Mario Geiger and Maurice Weiler,
A General Theory of Equivariant CNNs on Homogeneous Spaces,
Conference on Neural Information Processing Systems (NeurIPS), 2019

Steerable CNNs

- Standard *convolution* with G*-steerable filter* K guarantees also G **equivariance**

$$K: \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}}$$

$$K(x) = \rho_{out}(g) K(g^{-1}.x) \rho_{in}(g)^T$$

Steerability Constraint

Q: How do we parameterize G-steerable filters?

Equivariant Non-Linearities

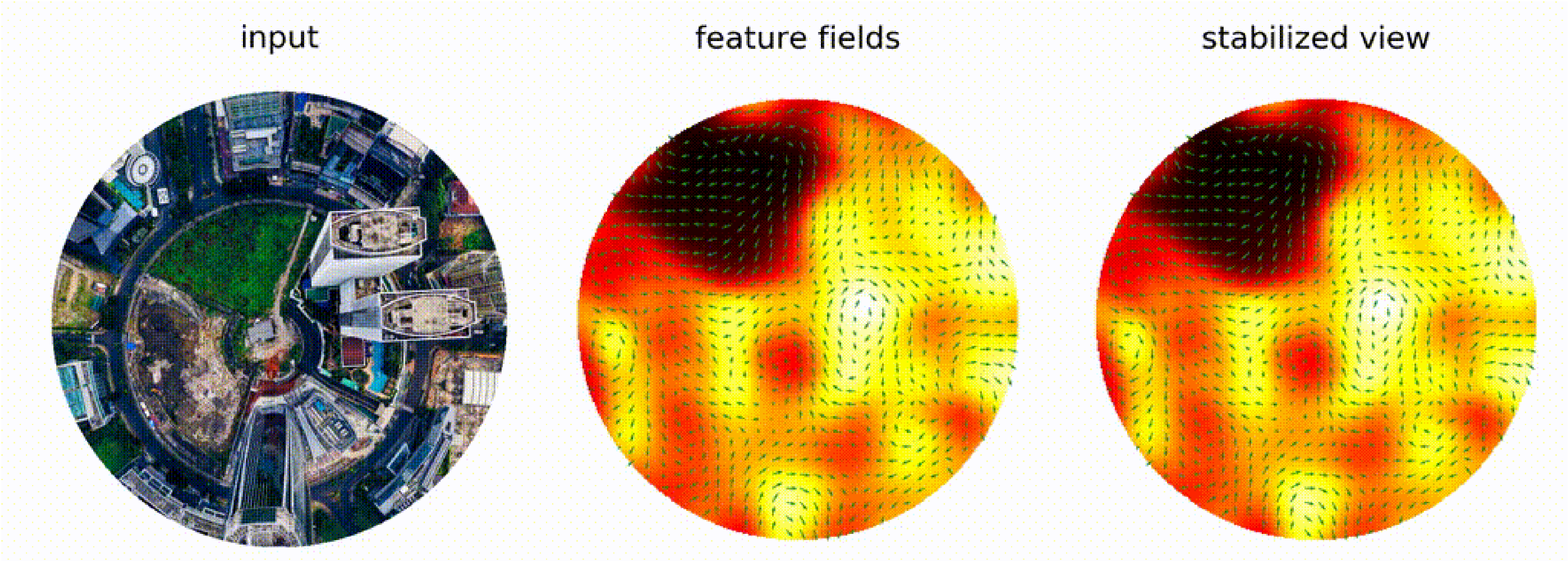
- Let the intermediate feature $f: \mathbb{R}^n \to \mathbb{R}^d$ transform under $\rho: G \to O(d)$
- The quantity $|f(x)|_2^2 \in \mathbb{R}^+$ is invariant
  - **norm non-linearity:** $f(x) \mapsto \sigma(|f(x)|_2^2)\, f(x)$ (Worrall et al., 2017)
  - **gated non-linearity:** $f(x), f_g(x) \mapsto \sigma\left(f_g(x)\right) f(x)$ (Weiler et al., 2018)
    - where $f_g(x)$ is another, invariant, feature field transforming under $\rho(g) = 1$
- Can also use other quadratic invariants:
  - **tensor-product**: $f(x) \mapsto f(x) \otimes f(x) \in \mathbb{R}^{d^2}$ (Kondor et al., 2018)
  - output transforms under $\rho_{out} = \rho \otimes \rho$

$$(x, y, z) \otimes (x, y, z) = vec\begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}$$

- Fourier-based pointwise non-linearities (imitate GCNN)
  - feature vector $f(x) \in \mathbb{R}^d$ represents a bandlimited signal in $L^2(G)$
  - Compose: (discrete) *Fourier Transform* $\circ$ σ $\circ$ (discrete) *Inverse Fourier Transform*
  - Band-limit + sufficient samples to control *reconstruction error*

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *Conference on Computer Vision and Pattern Recognition (CVPR), 2017.*
Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S. Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. *Conference on Neural Information Processing Systems (NeurIPS), 2018*
Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-gordan nets: a fully Fourier space spherical convolutional neural network. Advances in Neural Information Processing Systems (NeurIPS), 2018.
Nadav Dym, and Haggai Maron. On the Universality of Rotation Equivariant Point Cloud Networks. International Conference on Learning Representations (ICML). 2020.

Outcome: equivariance to continuous rotations



Source: e2cnn

Agenda

From Group Convolution to Steerable Filters

Steerable Fields and Representation Theory

Steerable CNNs

Hard Priors: solving the exact kernel constraint

Soft Priors: learnable kernel constraint

Solving the steerability constraint

- Standard *convolution* with G*-steerable filter* K guarantees also G **equivariance**

$$K: \mathbb{R}^n \rightarrow \mathbb{R}^{d_{out} \times d_{in}}$$

$$K(x) = \rho_{out}(g)K(g^{-1}.x)\rho_{in}(g)^T$$

Steerability Constraint

Q: How do we parameterize G-steerable filters?

Solving the steerability constraint: **Overall Strategy**

1. Linear projection Π: space of *unconstrained kernels* $\longmapsto$ space of *equivariant kernels*
2. Pick a convenient basis for the **domain** of Π: space of *unconstrained kernels*
3. Project to find a basis for the **image** of Π: space of *equivariant kernels*

$$K(x) = \rho_{out}(g) K(g^{-1}.x) \rho_{in}(g)^T$$

Solving the steerability constraint: **Projection**

- Linear projection Π: space of *unconstrained kernels* $\quad\longmapsto\quad$ space of *equivariant kernels*

$$K': \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}} \qquad \overset{\Pi}{\longmapsto} \qquad K = \Pi[K']: \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}}$$

$$K(x) = \rho_{out}(g) K(g^{-1}.x) \rho_{in}(g)^T$$

Solving the steerability constraint: **Reynolds Operator**

- Linear projection Π: space of *unconstrained kernels* $\longmapsto$ space of *equivariant kernels*

$$K': \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}} \quad \overset{\Pi}{\longmapsto} \quad K = \Pi[K']: \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}}$$

$$K(x) = \rho_{out}(g) K(g^{-1}.x) \rho_{in}(g)^T$$

$$\Pi[K'](x) = \int_G \rho_{out}(g) K'(g^{-1}.x)\, \rho_{in}(g)^{\mathrm{T}}\, dg$$

Solving the steerability constraint: Kernel Vectorization

- Consider the vectorized kernel $\kappa(x) = vec(K(x))$
  - Column-wise vectorization of the matrix $K(x)$

Identity

$$vec(ABC^T) = (C \otimes A)vec(B)$$

Kronecker product

$$A \otimes B = \begin{bmatrix} a_{11} \cdot B & \cdots & a_{1n} \cdot B \\ \vdots & \ddots & \vdots \\ a_{n1} \cdot B & \cdots & a_{nn} \cdot B \end{bmatrix}$$

$$\rho_{in} \otimes \rho_{out} : G \to \mathbb{R}^{(d_{out} \cdot d_{in}) \times (d_{out} \cdot d_{in})}$$

Vectorized Steerability Constraint

$$\kappa : \mathbb{R}^n \to \mathbb{R}^{d_{out} \cdot d_{in}}$$

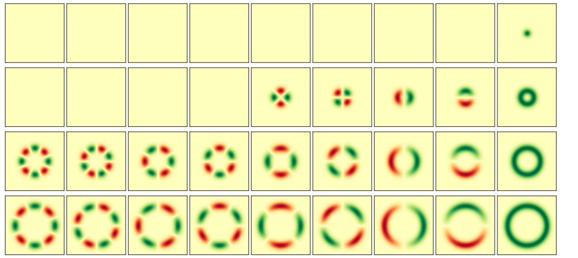
$$\kappa(x) = [(\rho_{in} \otimes \rho_{out})(g)]\kappa(g^{-1}.x)$$

$$\Pi[\kappa'](x) = \int_G (\rho_{in} \otimes \rho_{out})(g)\kappa'(g^{-1}.x)\, dg$$

Solving the steerability constraint: Steerable Basis

- Assume a G-steerable basis $\mathcal{B} = \{Y_j^k : \mathbb{R}^n \to \mathbb{R}^{d_j} \mid \psi_j \in \hat{G}, k\}$ for $L^2(\mathbb{R}^n)$ (Freeman & Adelson, 1991)

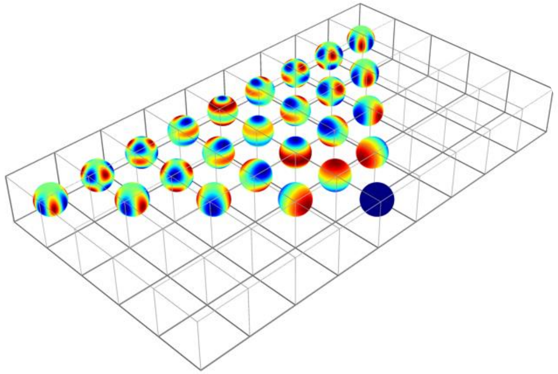
$$Y_j^k(g^{-1}.x) = \psi_j(g)^T Y_j^k(x)$$



$$\psi_j(\theta) = \begin{bmatrix} \cos j\theta & -\sin j\theta \\ \sin j\theta & \cos j\theta \end{bmatrix}$$

- Expand *unconstrained kernel* with parameter matrices $W_{j,k} \in \mathbb{R}^{d_{out} \cdot d_{in} \times d_j}$

$$\kappa'(x) = \sum_{j,k} W_{j,k} Y_j^k(x)$$



$\psi_j(g) = D^j(\alpha, \beta, \gamma) \in \mathbb{R}^{2j+1 \times 2j+1}$
Wigner D-matrix

William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1991

Solving the steerability constraint: Expand Basis

- Expand *unconstrained kernel* with parameter matrices $W_{j,k} \in \mathbb{R}^{d_{out} \cdot d_{in} \times d_j}$

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\rho_{in} \otimes \rho_{out})(g) \kappa'(g^{-1}.x)\, dg$$

$$= \int_G (\rho_{in} \otimes \rho_{out})(g) \sum_{j,k} W_{j,k} Y_j^k (g^{-1}.x)\, dg$$

Solving the steerability constraint: Expand Basis

- Expand *unconstrained kernel* with parameter matrices $W_{j,k} \in \mathbb{R}^{d_{out} \cdot d_{in} \times d_j}$

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\rho_{in} \otimes \rho_{out})(g)\kappa'(g^{-1}.x)\, dg$$

$$= \int_G (\rho_{in} \otimes \rho_{out})(g) \sum_{j,k} W_{j,k}\psi_j(g)^T Y_j^k(x)\, dg$$

Solving the steerability constraint: Expand Basis

- Expand *unconstrained kernel* with parameter matrices $W_{j,k} \in \mathbb{R}^{d_{out} \cdot d_{in} \times d_j}$

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\rho_{in} \otimes \rho_{out})(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} \left[\int_G (\rho_{in} \otimes \rho_{out})(g) W_{j,k}\, \psi_j(g)^T dg \right] Y_j^k(x)$$

Solving the steerability constraint: Expand Basis

- Expand *unconstrained kernel* with parameter matrices $W_{j,k} \in \mathbb{R}^{d_{out} \cdot d_{in} \times d_j}$

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\rho_{in} \otimes \rho_{out})(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

$$\left[\int_G (\psi_j \otimes \rho_{in} \otimes \rho_{out})(g)\, dg\right] vec(W_{j,k})$$

Identity

$vec(ABC^T) = (C \otimes A)vec(B)$

Solving the steerability constraint: Assume Irreps

- W.l.o.g. assume input and output representations are *irreducible representations*
  - That is $\rho_{out} = \psi_J$ and $\rho_{in} = \psi_l$

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g) \kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\ \ ]\, Y_j^k(x)$$

$$\left[\int_G (\psi_j \otimes \psi_l \otimes \psi_J)(g)\, dg\right] vec(W_{j,k})$$

Peter-Weyl Theorem

Decompose $\psi_j \otimes \psi_l \otimes \psi_J$

Solving the steerability constraint: Decompose Tensor Products

- Tensor products can be decomposed as a direct sum of irreps via Clebsh-Gordan transform

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\quad]\, Y_j^k(x)$$

$$\left[\int_G (\psi_j \otimes \psi_l \otimes \psi_J)(g)\, dg\right] vec(W_{j,k})$$

Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations*, 2020

Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build $E(N)$-equivariant steerable CNNs. *International Conference on Learning Representations. 2021*

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Peter-Weyl Theorem

Decompose $\psi_j \otimes \psi_l \otimes \psi_J$

Solving the steerability constraint: Decompose Tensor Products

- Tensor products can be decomposed as a direct sum of irreps via Clebsh-Gordan transform

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\ \ ]\, Y_j^k(x)$$

$$\left[\int_G Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \psi_i(g) \right) Q_{jlJ}\, dg \right] vec(W_{j,k})$$

Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations*, 2020
Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build $E(N)$-equivariant steerable CNNs. *International Conference on Learning Representations. 2021*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Solving the steerability constraint: Decompose Tensor Products

- Tensor products can be decomposed as a direct sum of irreps via Clebsh-Gordan transform

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

$$Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \left[\int_G \psi_i(g)\, dg \right] \right) Q_{jlJ}\, vec(W_{j,k})$$

Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations*, 2020

Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build $E(N)$-equivariant steerable CNNs. *International Conference on Learning Representations. 2021*

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Solving the steerability constraint: Recall Fourier Transform

- The matrix coefficients of the *irreducible representations* form an *orthogonal basis*

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

$$Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \left[\int_G \psi_i(g) \underbrace{\psi_0(g)}_{=1} \, dg \right] \right) Q_{jlJ}\, vec(W_{j,k})$$

Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations*, 2020
Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build $E(N)$-equivariant steerable CNNs. *International Conference on Learning Representations. 2021*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Peter-Weyl Theorem
Orthogonality matrix coefficients of the irreps

Solving the steerability constraint: Recall Fourier Transform

- The matrix coefficients of the *irreducible representations* form an *orthogonal basis*

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\quad]\, Y_j^k(x)$$

$$Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \delta_{i=0} \right) Q_{jlJ}\, vec(W_{j,k})$$

Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations*, 2020
Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build $E(N)$-equivariant steerable CNNs. *International Conference on Learning Representations. 2021*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Solving the steerability constraint: Sparse Subset of Weights

- The matrix coefficients of the *irreducible representations* form an *orthogonal basis*

$$\kappa(x) = \Pi[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\, dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

Π_{jlJ}

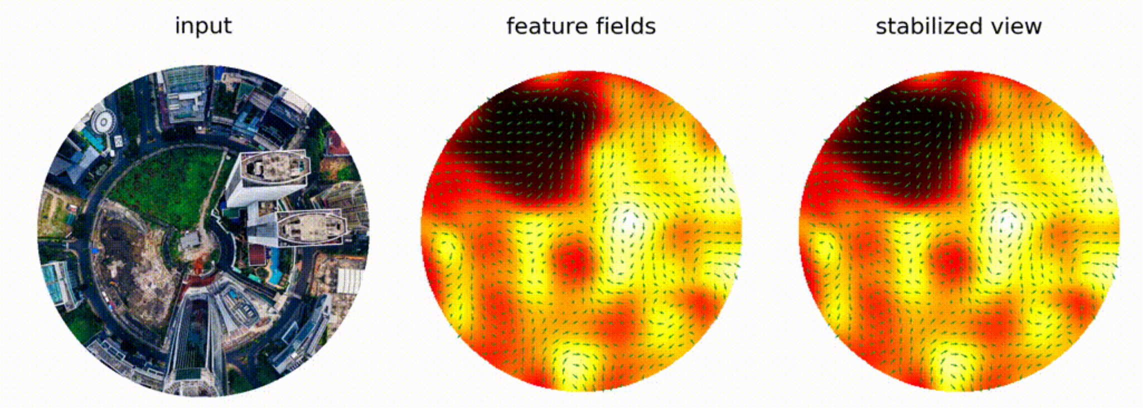
$$Q_{jlJ}^T P_{jlJ} Q_{jlJ} vec(W_{j,k})$$

$$P_{jlJ} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

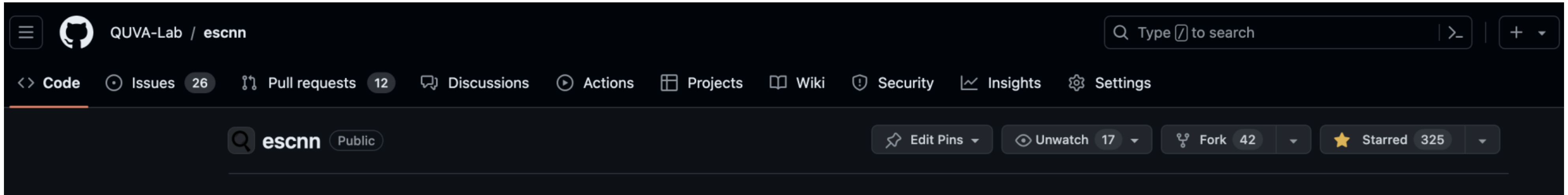
Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations*, 2020
Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build $E(N)$-equivariant steerable CNNs. *International Conference on Learning Representations. 2021*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Result

- Complete theoretical description of the space of G-steerable filters
  - For any compact G and any transformation laws ρ_{in}, ρ_{out}
- Algorithm to explicitly construct the steerable convolution layers
- General implementation in the form of a *PyTorch library:*



github.com/QUVA-Lab/escnn



General Program to implement G-equivariance: 2D images

2D rotational symmetries

$$(\mathbb{R}^2, +) \rtimes G < E(2)$$

	group	representation		nonlinearity	invariant map	citation	MNIST O(2)	MNIST rot	MNIST 12k
1	$\{e\}$	(conventional CNN)		ELU	-	-	5.53 ± 0.20	2.87 ± 0.09	0.91 ± 0.06
2	C_1					[7, 9]	5.19 ± 0.08	2.48 ± 0.13	0.82 ± 0.01
3	C_2					[7, 9]	3.29 ± 0.07	1.32 ± 0.02	0.87 ± 0.04
4	C_3					-	2.87 ± 0.04	1.19 ± 0.06	0.80 ± 0.03
5	C_4					[6, 1, 7, 9, 10	2.40 ± 0.05	1.02 ± 0.03	0.99 ± 0.03
6	C_6	regular	ρ_{reg}	ELU	G-pooling	[8]	2.08 ± 0.03	0.89 ± 0.03	0.84 ± 0.02
7	C_8					[7, 9]	1.96 ± 0.04	0.84 ± 0.02	0.89 ± 0.03
8	C_{12}					[7]	1.95 ± 0.07	0.80 ± 0.03	0.89 ± 0.03
9	C_{16}					[7, 9]	1.93 ± 0.04	0.82 ± 0.02	0.95 ± 0.04
10	C_{20}					[7]	1.95 ± 0.05	0.83 ± 0.05	0.94 ± 0.06
11	C_4		$5\rho_{reg} \oplus 2\rho_{quot}^{C_4/C_2} \oplus 3\psi_0$			[1]	2.43 ± 0.05	1.03 ± 0.05	1.01 ± 0.03
12	C_8		$5\rho_{reg} \oplus 2\rho_{quot}^{C_6/C_2} \oplus 2\rho_{quot}^{C_6/C_3} \oplus 2\psi_0$			-	2.03 ± 0.05	0.84 ± 0.05	0.91 ± 0.02
13	C_{12}	quotient	$5\rho_{reg} \oplus 2\rho_{quot}^{C_8/C_2} \oplus 2\rho_{quot}^{C_8/C_4} \oplus 2\psi_0$			-	2.04 ± 0.04	0.81 ± 0.02	0.95 ± 0.02
14	C_{16}		$5\rho_{reg} \oplus 2\rho_{quot}^{C_{12}/C_2} \oplus 2\rho_{quot}^{C_{12}/C_4} \oplus 3\psi_0$			-	2.00 ± 0.01	0.86 ± 0.04	0.98 ± 0.04
15	C_{20}		$5\rho_{reg} \oplus 2\rho_{quot}^{C_{16}/C_2} \oplus 2\rho_{quot}^{C_{16}/C_4} \oplus 4\psi_0$			-	2.01 ± 0.05	0.83 ± 0.03	0.96 ± 0.04
16		regular/scalar	$\psi_0 \xrightarrow{\text{conv}} \rho_{reg} \xrightarrow{G\text{-pool}} \psi_0$	ELU, G-pooling		[6, 25]	2.02 ± 0.02	0.90 ± 0.03	0.93 ± 0.04
17	C_{16}	regular/vector	$\psi_1 \xrightarrow{\text{conv}} \rho_{reg} \xrightarrow{\text{vector pool}} \psi_1$	vector field		[13, 26]	2.12 ± 0.02	1.07 ± 0.03	0.78 ± 0.03
18		mixed vector	$\rho_{reg} \oplus \psi_1 \xrightarrow{\text{conv}} 2\rho_{reg} \xrightarrow[\text{pool}]{\text{vector}} \rho_{reg} \oplus \psi_1$	ELU, vector field		-	1.87 ± 0.03	0.83 ± 0.02	0.63 ± 0.02
19	D_1					-	3.40 ± 0.07	3.44 ± 0.10	0.98 ± 0.03
20	D_2					-	2.42 ± 0.07	2.39 ± 0.04	1.05 ± 0.03
21	D_3					-	2.17 ± 0.06	2.15 ± 0.05	0.94 ± 0.02
22	D_4					[6, 1, 27]	1.88 ± 0.04	1.87 ± 0.04	1.69 ± 0.03
23	D_6	regular	ρ_{reg}	ELU	G-pooling	[8]	1.77 ± 0.06	1.77 ± 0.04	1.00 ± 0.03
24	D_8					-	1.68 ± 0.06	1.73 ± 0.03	1.64 ± 0.02
25	D_{12}					-	1.66 ± 0.05	1.65 ± 0.05	1.67 ± 0.01
26	D_{16}					-	1.62 ± 0.04	1.65 ± 0.02	1.68 ± 0.04
27	D_{20}					-	1.64 ± 0.06	1.62 ± 0.05	1.69 ± 0.03
28	D_{16}	regular/scalar	$\psi_{0,0} \xrightarrow{\text{conv}} \rho_{reg} \xrightarrow{G\text{-pool}} \psi_{0,0}$	ELU, G-pooling		-	1.92 ± 0.03	1.88 ± 0.07	1.74 ± 0.04
29		irreps ≤ 1	$\bigoplus_{i=0}^{1} \psi_i$			-	2.98 ± 0.04	1.38 ± 0.09	1.29 ± 0.05
30		irreps ≤ 3	$\bigoplus_{i=0}^{3} \psi_i$			-	3.02 ± 0.18	1.38 ± 0.09	1.27 ± 0.03
31		irreps ≤ 5	$\bigoplus_{i=0}^{5} \psi_i$			-	3.24 ± 0.05	1.44 ± 0.10	1.36 ± 0.04
32		irreps ≤ 7	$\bigoplus_{i=0}^{7} \psi_i$	ELU, norm-ReLU	conv2triv	-	3.30 ± 0.11	1.51 ± 0.10	1.40 ± 0.07
33		$\mathbb{C}$-irreps ≤ 1	$\bigoplus_{i=0}^{1} \psi_i^{\mathbb{C}}$			[12]	3.39 ± 0.10	1.47 ± 0.06	1.42 ± 0.04
34		$\mathbb{C}$-irreps ≤ 3	$\bigoplus_{i=0}^{3} \psi_i^{\mathbb{C}}$			[12]	3.48 ± 0.16	1.51 ± 0.05	1.53 ± 0.07
35		$\mathbb{C}$-irreps ≤ 5	$\bigoplus_{i=0}^{5} \psi_i^{\mathbb{C}}$			-	3.59 ± 0.08	1.59 ± 0.05	1.55 ± 0.06
36	SO(2)	$\mathbb{C}$-irreps ≤ 7	$\bigoplus_{i=0}^{7} \psi_i^{\mathbb{C}}$			-	3.64 ± 0.12	1.61 ± 0.06	1.62 ± 0.03
37				ELU, squash		-	3.10 ± 0.09	1.41 ± 0.04	1.46 ± 0.05
38				ELU, norm-ReLU		-	3.23 ± 0.08	1.38 ± 0.08	1.33 ± 0.03
39				ELU, shared norm-ReLU	norm	-	2.88 ± 0.11	1.15 ± 0.06	1.18 ± 0.03
40		irreps ≤ 3	$\bigoplus_{i=0}^{3} \psi_i$	shared norm-ReLU		-	3.61 ± 0.09	1.57 ± 0.05	1.88 ± 0.05
41				ELU, gate	conv2triv	-	2.37 ± 0.06	1.09 ± 0.03	1.10 ± 0.02
42				ELU, shared gate		-	2.33 ± 0.06	1.11 ± 0.03	1.12 ± 0.04
43				ELU, gate	norm	-	2.23 ± 0.09	1.04 ± 0.04	1.05 ± 0.06
44				ELU, shared gate		-	2.20 ± 0.06	1.01 ± 0.03	1.03 ± 0.03
45		irreps $= 0$	$\psi_{0,0}$	ELU	-	-	5.46 ± 0.46	5.21 ± 0.29	3.98 ± 0.04
46		irreps ≤ 1	$\psi_{0,0} \oplus \psi_{1,0} \oplus 2\psi_{1,1}$			-	3.31 ± 0.17	3.37 ± 0.18	3.05 ± 0.09
47		irreps ≤ 3	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{3} 2\psi_{1,i}$	ELU, norm-ReLU	O(2)-conv2triv	-	3.42 ± 0.03	3.41 ± 0.10	3.86 ± 0.09
48		irreps ≤ 5	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{5} 2\psi_{1,i}$			-	3.59 ± 0.13	3.78 ± 0.31	4.17 ± 0.15
49		irreps ≤ 7	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{7} 2\psi_{1,i}$			-	3.84 ± 0.25	3.90 ± 0.18	4.57 ± 0.27
50		Ind-irreps ≤ 1	$\text{Ind}\,\psi_0^{SO(2)} \oplus \text{Ind}\,\psi_1^{SO(2)}$			-	2.72 ± 0.05	2.70 ± 0.11	2.39 ± 0.07
51	O(2)	Ind-irreps ≤ 3	$\text{Ind}\,\psi_0^{SO(2)} \bigoplus_{i=1}^{3} \text{Ind}\,\psi_i^{SO(2)}$	ELU, Ind norm-ReLU	Ind-conv2triv	-	2.66 ± 0.07	2.65 ± 0.12	2.25 ± 0.06
52		Ind-irreps ≤ 5	$\text{Ind}\,\psi_0^{SO(2)} \bigoplus_{i=1}^{5} \text{Ind}\,\psi_i^{SO(2)}$			-	2.71 ± 0.11	2.84 ± 0.10	2.39 ± 0.09
53		Ind-irreps ≤ 7	$\text{Ind}\,\psi_0^{SO(2)} \bigoplus_{i=1}^{7} \text{Ind}\,\psi_i^{SO(2)}$			-	2.80 ± 0.12	2.85 ± 0.06	2.25 ± 0.08
54		irreps ≤ 3	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{3} 2\psi_{1,i}$	ELU, gate	O(2)-conv2triv	-	2.39 ± 0.05	2.38 ± 0.07	2.28 ± 0.07
55					norm	-	2.21 ± 0.09	2.24 ± 0.06	2.15 ± 0.03
56		Ind-irreps ≤ 3	$\text{Ind}\,\psi_0^{SO(2)} \bigoplus_{i=1}^{3} \text{Ind}\,\psi_i^{SO(2)}$	ELU, Ind gate	Ind-conv2triv	-	2.13 ± 0.04	2.09 ± 0.05	2.05 ± 0.05
57					Ind-norm	-	1.96 ± 0.06	1.95 ± 0.05	1.85 ± 0.07

MNIST Variations

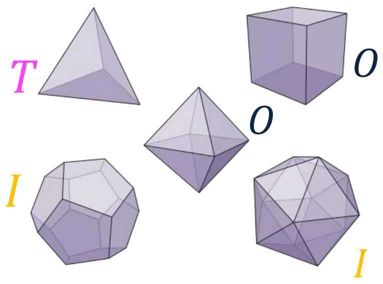
Maurice Weiler* and Gabriele Cesa*.
General E(2)-Equivariant Steerable CNNs,
Neural Information Processing Systems (NeurIPS), 2019

General Program to implement G-equivariance: 3D voxel data

Table 1: Rotated ModelNet10 (O(3) symmetry). $*$ indicates wider models to fix the computational cost.

G	Description	Accuracy
$\{e\}$	Conventional CNN	82.5 ± 1.4
SO(2)	Axial Symmetry	86.9 ± 1.9
$\mathrm{SO}(2) \rtimes \mathrm{F} \cong \mathrm{O}(2)$	Dihedral Symmetry	87.5 ± 0.7
$\mathrm{SO}(2) \rtimes \mathrm{M} \cong \mathrm{O}(2)$	Conical Symmetry	88.5 ± 0.8
$\mathrm{Inv} \times \mathrm{SO}(2)$	Cylindrical Symmetry	86.8 ± 0.7
$\mathrm{Inv} \times \mathrm{SO}(2) \rtimes \mathrm{F}$	Full Cylindrical Symmetry	87.0 ± 1.0
O	Octahedral Symmetry (Winkels & Cohen, 2018)	89.7 ± 0.6
I	Icosahedral Symmetry	90.0 ± 0.6
I	Icosahedral Symmetry (finite orbits basis)	88.2 ± 1.0
SO(3)	Chiral (Tensor product) (Anderson et al., 2019)	86.3 ± 1.0
SO(3)	Chiral (Gated non-linearity)(Weiler et al., 2018b)	88.8 ± 1.2
SO(3)	Chiral (Regular, $\lvert\mathcal{G}\rvert = 96$)	89.1 ± 1.2
SO(3)	Chiral (Regular, $\lvert\mathcal{G}\rvert = 192$)*	89.4 ± 1.4
SO(3)	Chiral (Quotient $\mathcal{S}^2 = \mathrm{SO}(3)/\mathrm{SO}(2)$, $\lvert\mathcal{X}\rvert = 30$)	89.5 ± 1.0
O(3)	Achiral (Regular, $\lvert\mathcal{G}\rvert = 120$)	89.2 ± 0.6
O(3)	Achiral (Regular, $\lvert\mathcal{G}\rvert = 144$)*	89.4 ± 0.7
O(3)	Achiral (Quotient $\mathrm{Inv} \times \mathcal{S}^2 = \mathrm{O}(3)/\mathrm{SO}(2)$, $\lvert\mathcal{X}\rvert = 60$)	88.6 ± 0.9

3D rotational symmetries



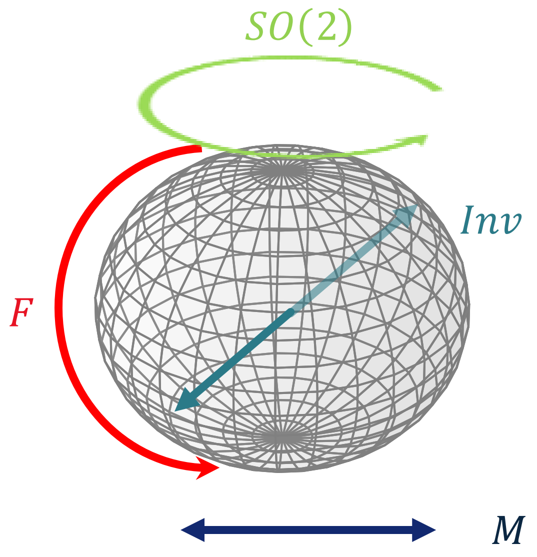


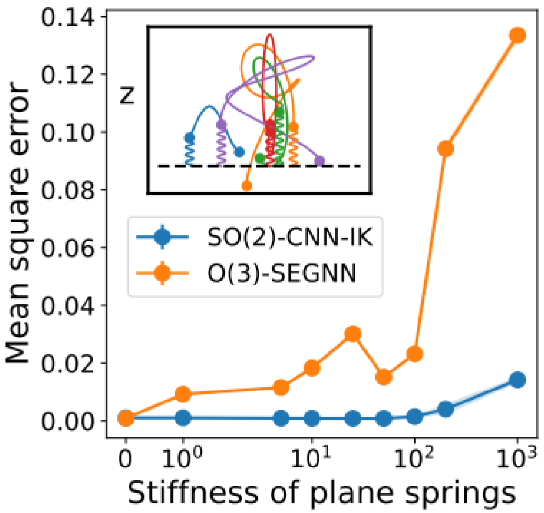
Table 2: ModelNet10 (O(2) symmetry)

G	Description	Accuracy
$\{e\}$	Conventional CNN	91.2 ± 0.5
SO(2)	Azimuthal Symmetry	91.9 ± 0.8
SO(3)	Chiral (Regular, $\lvert\mathcal{G}\rvert = 72$)	89.8 ± 0.6
O(2)	Full Azimuthal Symmetry	92.3 ± 0.4
O(3)	Achiral (Regular, $\lvert\mathcal{G}\rvert = 120$)	89.9 ± 1.0
$C_2 \rtimes \mathrm{F}$	Klein Group (dihedral symmetry)	91.0 ± 0.6
	VOXNet (Maturana & Scherer, 2015)	92.0
$C_2 \rtimes \mathrm{F}$	Klein Group (Worrall & Brostow, 2018)	94.2

Axial rotational symmetries in 3D

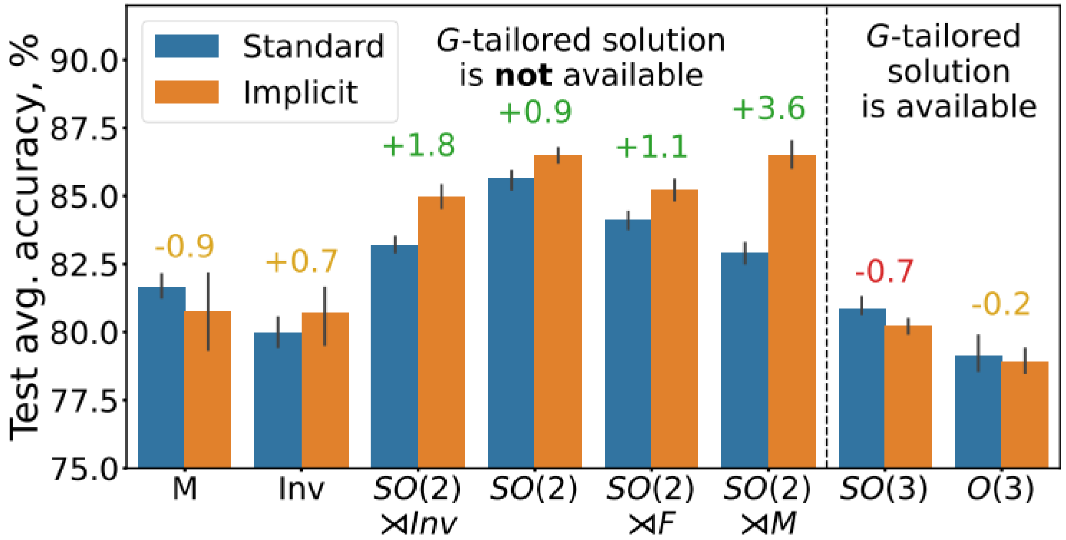
Gabriele Cesa, Leon Lang, Maurice Weiler,
A Program to build E(n)-Equivariant Steerable CNNs,
International Conference on Representation Learning, 2022

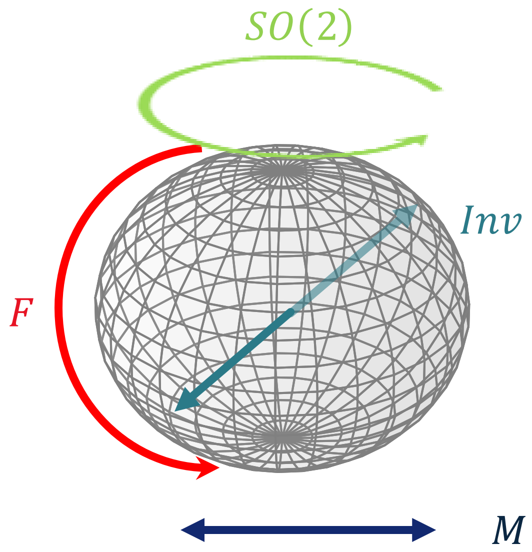
Beyond fixed Steerable Basis



N-body system
+ gravity

ModelNet40
Axial rotational symmetries in 3D





Learn G-equivariant MLP to parameterize G-steerable kernel

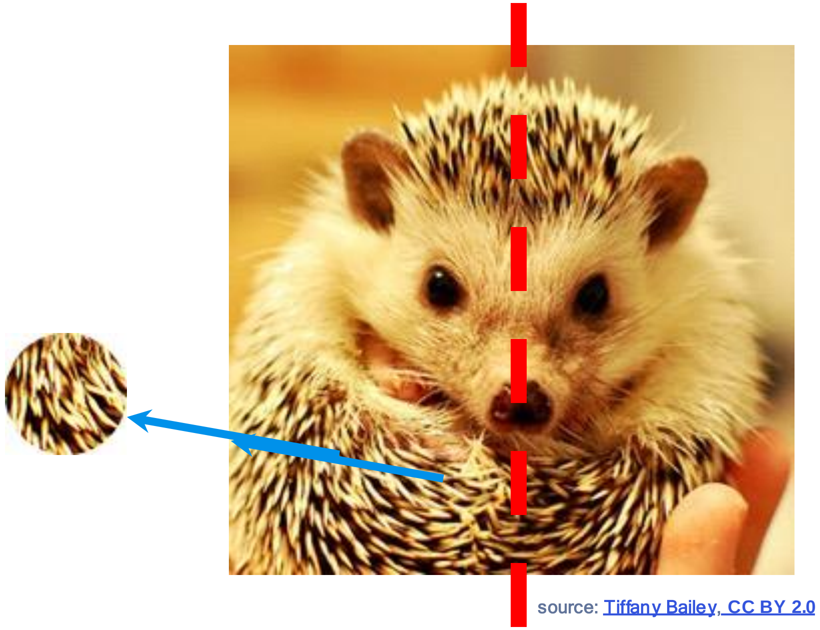
$$\kappa_\theta(g.x) = [(\rho_{in} \otimes \rho_{out})(g)]\kappa_\theta(x)$$

Maksim Zhdanov, Nico Hoffmann, Gabriele Cesa,
Implicit Convolutional Kernels for Steerable CNNs. *Neural Information Processing Systems, 2023*

Local Symmetries: Symmetries vary between features and scales

- Reflection symmetry in the class
- Rotational symmetry in the *local* patterns

source: MikeLynch, CC BY-SA 3.0



source: Tiffany Bailey, CC BY 2.0

Maurice Weiler* and Gabriele Cesa*.
General E(2)-Equivariant Steerable CNNs,
Neural Information Processing Systems (NeurIPS), 2019

Group Restriction

- Model the loss of symmetries at larger scales by relaxing the equivariance constraint at different depths:
  - exploit more symmetries in the first layers
  - restrict later to the symmetries of your output

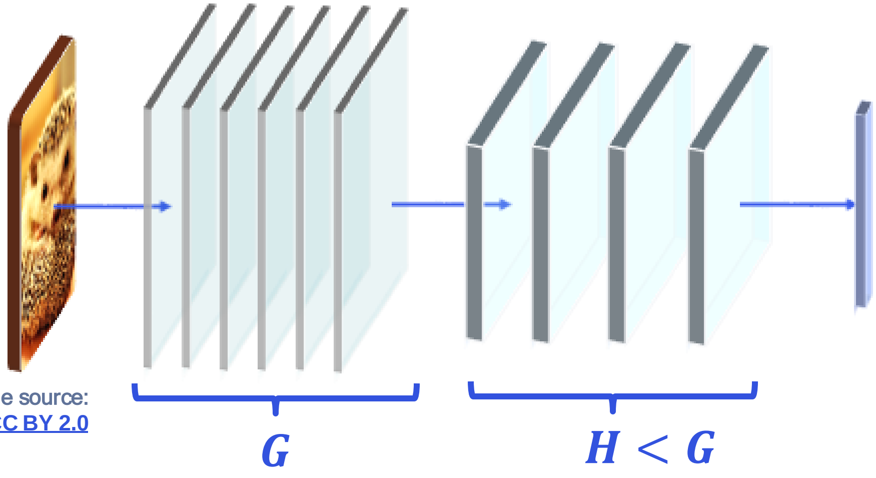


Image source:
Tiffany Bailey, CC BY 2.0

Maurice Weiler* and Gabriele Cesa*.
General E(2)-Equivariant Steerable CNNs,
Neural Information Processing Systems (NeurIPS), 2019

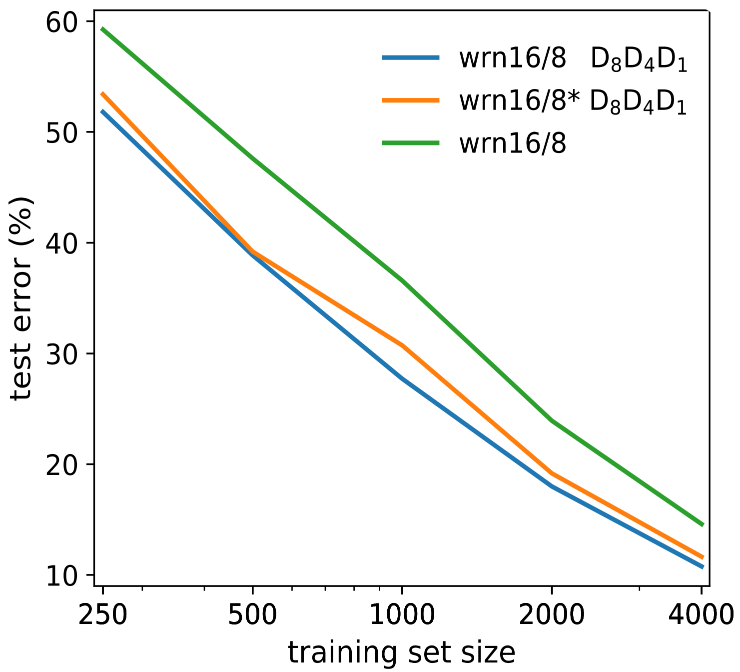
Experiments on Natural Images

model			CIFAR-10	CIFAR-100
wrn28/10	[30]		3.87	18.80
wrn28/10	$D_1\ D_1\ D_1$		3.36 ± 0.08	17.97 ± 0.11
wrn28/10*	$D_8\ D_4\ D_1$		3.32 ± 0.10	17.42 ± 0.38
wrn28/10	$C_8\ C_4\ C_1$		3.20 ± 0.04	16.47 ± 0.22
wrn28/10	$D_8\ D_4\ D_1$		3.13 ± 0.17	16.76 ± 0.40
wrn28/10	$D_8\ D_4\ D_4$		2.91 ± 0.13	16.22 ± 0.31
wrn28/10	[31]	AA	2.6 ± 0.1	17.1 ± 0.3
wrn28/10*	$D_8\ D_4\ D_1$	AA	2.39 ± 0.12	15.55 ± 0.13
wrn28/10	$D_8\ D_4\ D_1$	AA	2.05 ± 0.03	14.30 ± 0.09

AA = Auto Augment

model	group	#params	test error (%)
wrn16/8 [32]	-	11M	12.74 ± 0.23
wrn16/8*	$D_1\ D_1\ D_1$	5M	11.05 ± 0.45
wrn16/8	$D_1\ D_1\ D_1$	10M	11.17 ± 0.60
wrn16/8*	$D_8\ D_4\ D_1$	4.2M	10.57 ± 0.70
wrn16/8	$D_8\ D_4\ D_1$	12M	9.80 ± 0.40

STL - 10



Maurice Weiler* and Gabriele Cesa*.
General E(2)-Equivariant Steerable CNNs,
Neural Information Processing Systems (NeurIPS), 2019

Imperfect or Unknown symmetries

- *Group Restriction* : layer adapted to the symmetries manifested in the scale of its field of view
  - Still requires knowledge about these symmetries

- Can we *learn* the level of equivariance from data?

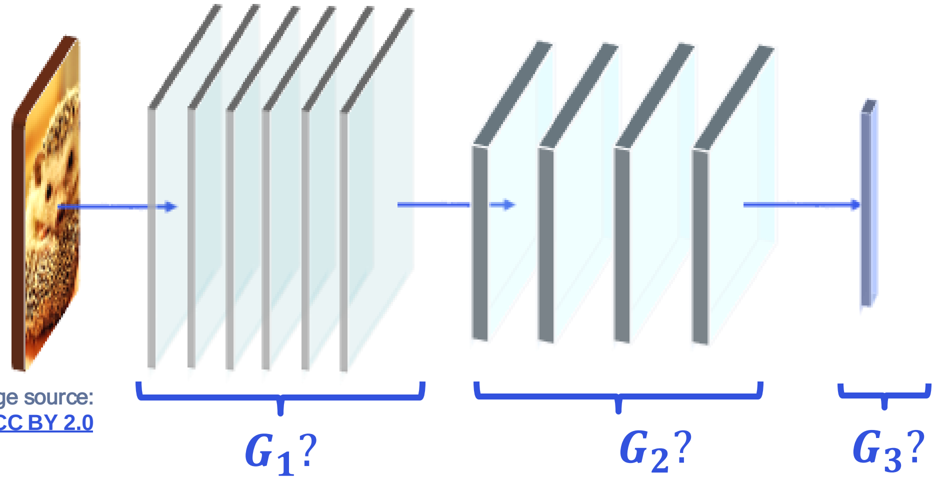


Image source:
Tiffany Bailey, CC BY 2.0

Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems (NeurIPS) 2021.*

van der Ouderaa, T., Romero, D. W., and van der Wilk, M. Relaxing equivariance constraints with non-stationary continuous filters. Advances in Neural Information Processing Systems (NeurIPS), 2022

Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics. International Conference on Machine Learning (ICML), 2022

Romero, D. W. and Lohit, S. Learning partial equivariances from data. Advances in Neural Information Processing Systems (NeurIPS), 2022

Agenda

From Group Convolution to Steerable Filters

Steerable Fields and Representation Theory

Steerable CNNs

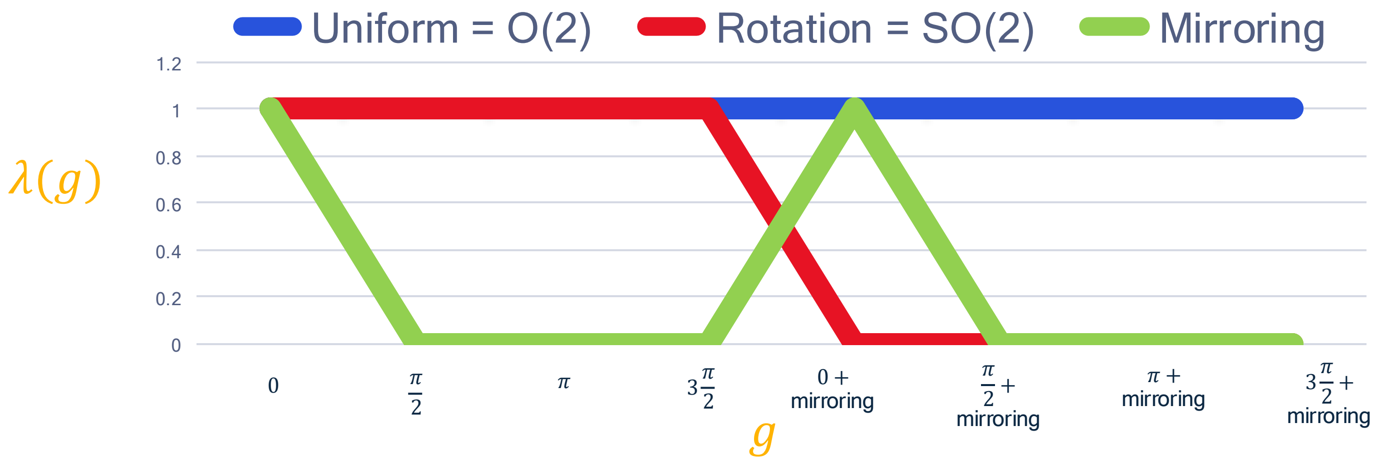
Hard Priors: solving the exact kernel constraint

Soft Priors: learnable kernel constraint

Learnable Soft Prior

Learn *probability distribution* λ over a large group G

- uniform: indicates equivariance over full group G
- Supported on subgroup $H \subset G$: indicates equivariance over full group G
- Low values outside subgroup $H \subset G$: indicates "soft / relaxed prior"



Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Learnable steerability constraint: Reynolds Operator

- Linear projection Π: space of *unconstrained kernels* $\longmapsto$ space of *equivariant kernels*

$$K': \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}} \quad \overset{\Pi}{\longmapsto} \quad K = \Pi[K']: \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}}$$

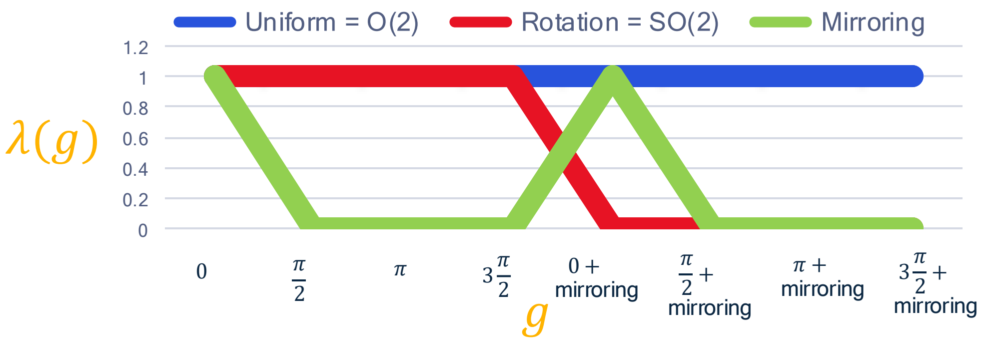
$$K(x) = \rho_{out}(g) K(g^{-1}.x) \rho_{in}(g)^T$$

$$\Pi[K'](x) = \int_G \rho_{out}(g) K'(g^{-1}.x)\, \rho_{in}(g)^{\mathrm{T}}\, dg$$

Learnable steerability constraint: Reynolds Operator

- Linear projection Π: space of *unconstrained kernels* $\longmapsto$ space of *equivariant kernels*

$$K' : \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}} \quad \overset{\Pi_\lambda}{\longmapsto} \quad K = \Pi_\lambda[K'] : \mathbb{R}^n \to \mathbb{R}^{d_{out} \times d_{in}}$$



$$\Pi_\lambda[K'](x) = \int_G \rho_{out}(g) K'(g^{-1}.x)\, \rho_{in}(g)^{\mathrm{T}} \lambda(g)\, dg$$

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Learnable steerability constraint: Repeat Previous Derivations

$$\kappa(x) = \Pi_\lambda[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\lambda(g)dg$$

$$= \sum_{j,k} unvec[\ ]\, Y_j^k(x)$$

$$Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \left[\int_G \psi_i(g)\lambda(g)\,dg \right] \right) Q_{jlJ}\ vec(W_{j,k})$$

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Learnable steerability constraint: Recall Fourier Transform

- The matrix coefficients of the *irreducible representations* form an *orthogonal basis*

$$\kappa(x) = \Pi_\lambda[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\lambda(g)dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

$$Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \left[\int_G \psi_i(g)\lambda(g)\, dg \right] \right) Q_{jlJ}\, vec(W_{j,k})$$

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Learnable steerability constraint: Recall Fourier Transform

- The matrix coefficients of the *irreducible representations* form an *orthogonal basis*

$$\kappa(x) = \Pi_\lambda[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\lambda(g)dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

$$Q_{jlJ}^T \left(\bigoplus_i \bigoplus_r^{[i(jlJ)]} \frac{\hat{\lambda}(\psi_i)}{\sqrt{d_i}} \right) Q_{jlJ}\, vec(W_{j,k})$$

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Learnable steerability constraint: Recall Fourier Transform

- The matrix coefficients of the *irreducible representations* form an *orthogonal basis*

$$\kappa(x) = \Pi_\lambda[\kappa'](x) = \int_G (\psi_l \otimes \psi_J)(g)\kappa'(g^{-1}.x)\lambda(g)dg$$

$$= \sum_{j,k} unvec[\quad] Y_j^k(x)$$

$\Pi_{\lambda jlJ}$

$$Q_{jlJ}^T P_{\lambda jlJ} Q_{jlJ}\, vec(W_{j,k})$$

$$P_{\lambda jlJ} = \begin{bmatrix} \frac{\hat{\lambda}(\psi_1)}{\sqrt{d_1}} & & & & \\ & 0 & & & \\ & & \frac{\hat{\lambda}(\psi_2)}{\sqrt{d_2}} & & \\ & & & \frac{\hat{\lambda}(\psi_0)}{\sqrt{d_0}} & \\ & & & & \ddots \\ & & & & & \frac{\hat{\lambda}(\psi_4)}{\sqrt{d_4}} \end{bmatrix}$$

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Side to Side comparison

$$\kappa(x) = \sum_{j,k} unvec\left[Q_{jlJ}^T \cdot Q_{jlJ}\, vec\left(W_{j,k}\right)\right] Y_j^k(x)$$

Π_{jlJ}

$$P_{jlJ} = \bigoplus_{i} \bigoplus_{r}^{[i(jlJ)]} \delta_{i=0}$$

$$P_{jlJ} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

$\Pi_{\lambda jlJ}$

$$P_{\lambda jlJ} = \bigoplus_{i} \bigoplus_{r}^{[i(jlJ)]} \frac{\hat{\lambda}(\psi_i)}{\sqrt{d_i}}$$

$$P_{\lambda jlJ} = \begin{bmatrix} \frac{\hat{\lambda}(\psi_1)}{\sqrt{d_1}} & & & & & \\ & 0 & & & & \\ & & \frac{\hat{\lambda}(\psi_2)}{\sqrt{d_2}} & & & \\ & & & \frac{\hat{\lambda}(\psi_0)}{\sqrt{d_0}} & & \\ & & & & \ddots & \\ & & & & & \frac{\hat{\lambda}(\psi_4)}{\sqrt{d_4}} \end{bmatrix}$$

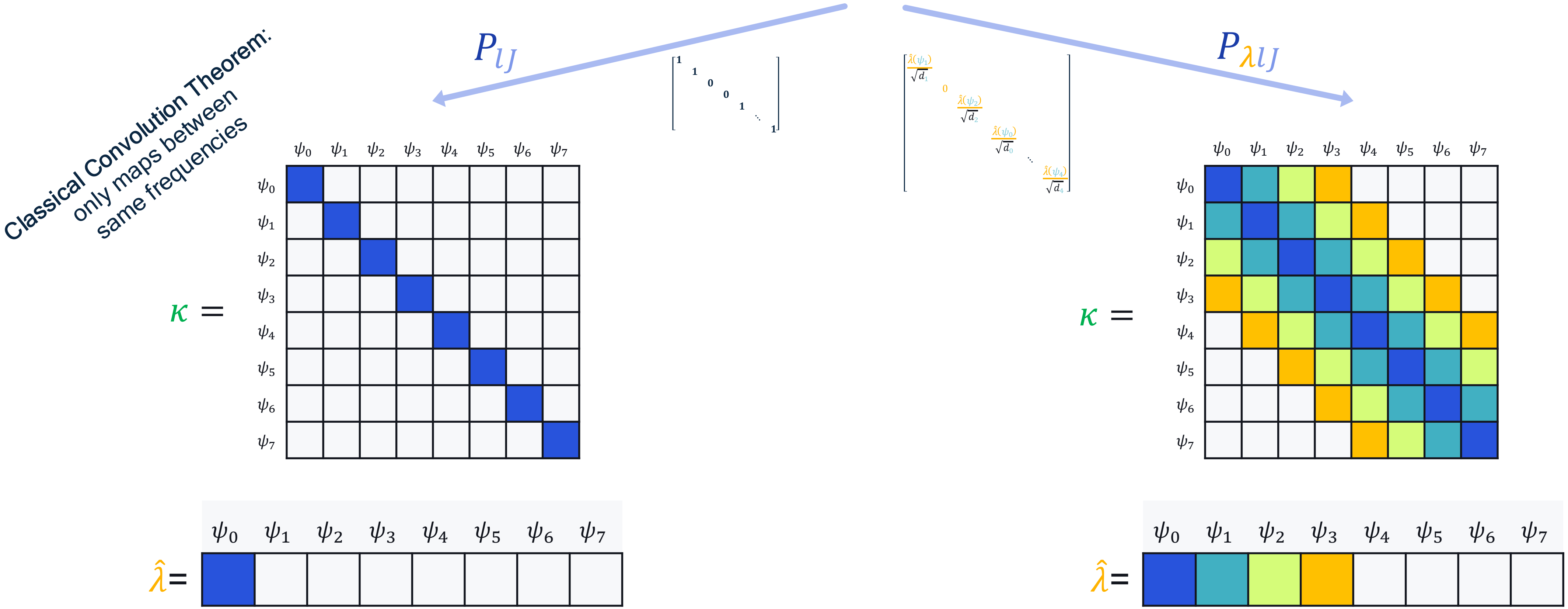
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Side to Side comparison: Simple MLP Setting

$$\psi_{\kappa}(\theta) = e^{i\kappa\theta}$$

- Consider MLP or 1x1 conv with $G = SO(2)$

$$\kappa = unvec\left[Q_{lJ}^T \cdot Q_{lJ}\, vec(W)\right]$$



Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Learnable steerability constraint: Implementation Details

- Everything is *differentiable* : can directly backpropagate to $\hat{\lambda}(\psi_i)$
- Normalise λ to a PDF:

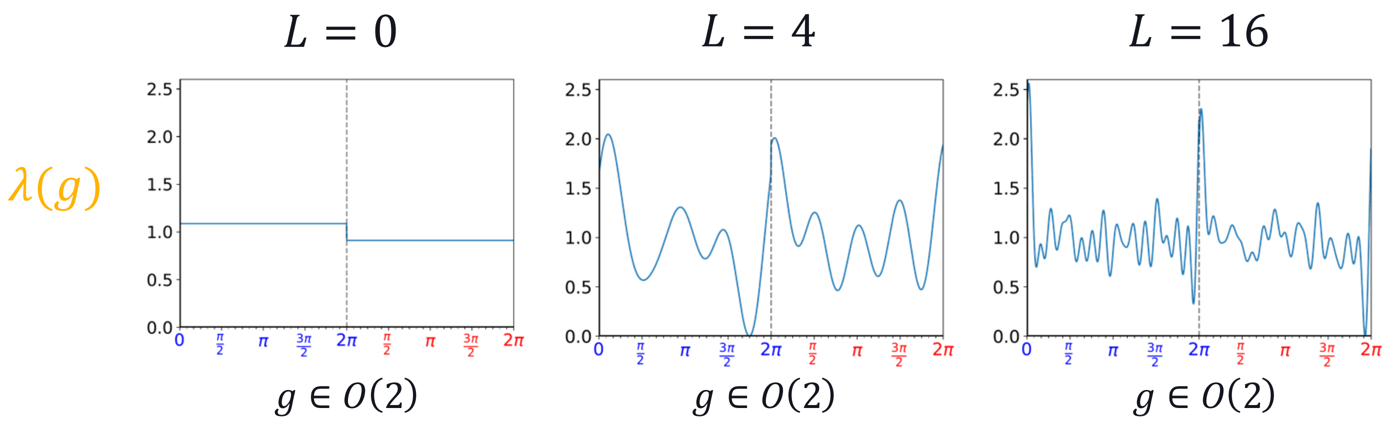
$$\hat{\lambda} = FT\left(\text{softmax}\left(IFT\left(\widehat{\lambda'}\right)\right)\right)$$

$\widehat{\lambda'}$ is the Fourier Transform of the *log-likelihood* function

- Initialize λ to uniform distribution

$$\hat{\lambda}(\psi_i) = \begin{cases} 1, & i = 0 \\ \mathbf{0}_{d_i \times d_i}, & i \neq 0 \end{cases}$$

- *Tunable* band-limit L on $\hat{\lambda}$ to regularise the likelihood and reduce parameters:



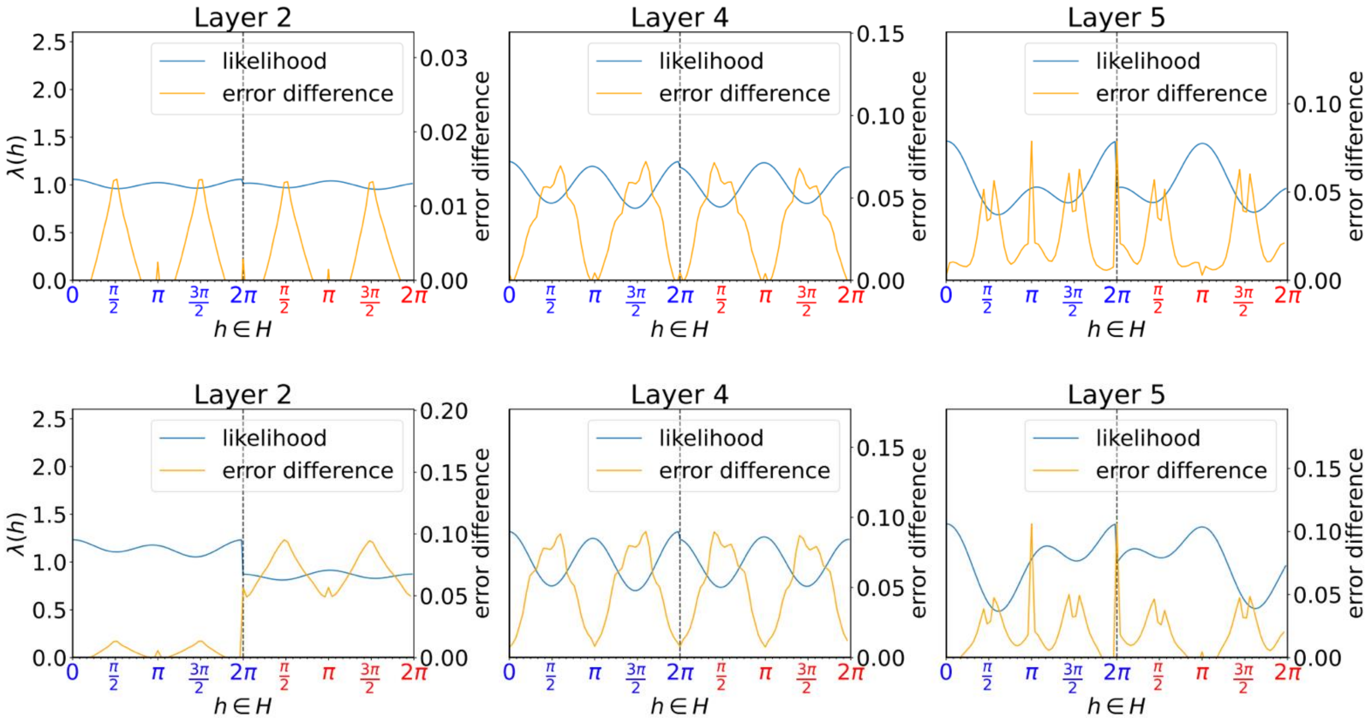
Learnable steerability constraint: Implementation Details

- Regularize subsequent layers with KL-divergence:

$$L_{KL} = \sum_{n}^{N-1} KL(\lambda_{n+1} || \text{detach}(\lambda_n))$$

With KL

Without KL

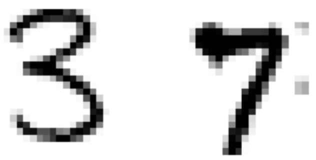


79

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Evaluation on Double-Digits MNIST

- Rectangular images containing 2 digits, independently transformed



(a) Original unaugmented number.

(b) Local vs global horizontal reflection.

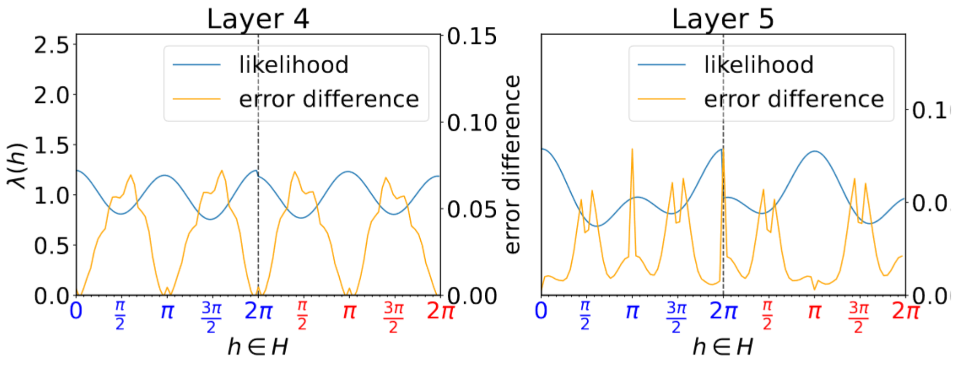
(c) Local vs global 90 degree rotation.

(d) Local vs global 180 degree rotation.

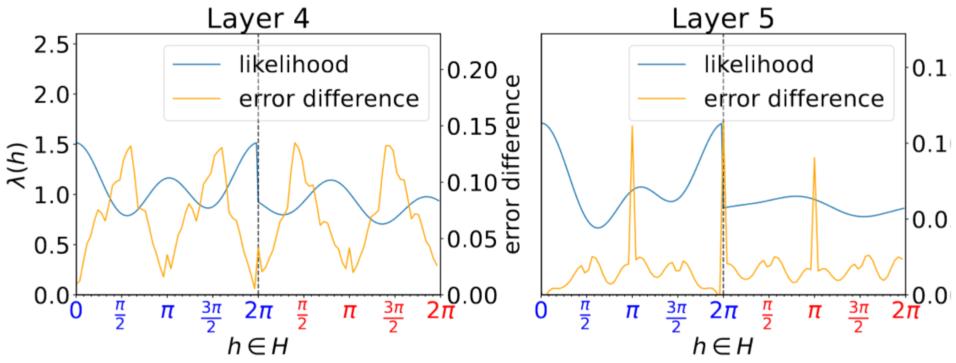
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Evaluation on Double-Digits MNIST

- Learnt Likelihood matches measured equivariance error



(a) $O(2)$ symmetries



(b) C_1 symmetries (i.e. no symmetries)

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Evaluation on Double-Digits MNIST

- Rectangular images containing 2 digits, independently transformed

Network Group	Partial Equivariance	Symmetries for individual digits C_1	C_4	$SO(2)$	D_1	D_4	$O(2)$
CNN	N/A	**0.962** (0.002)	0.868 (0.011)	0.807 (0.007)	0.919 (0.006)	0.711 (0.009)	0.649 (0.019)
C_4	None	0.933 (0.002)	0.484 (0.008)	0.459 (0.007)	0.893 (0.005)	0.427 (0.008)	0.405 (0.008)
	Restriction	0.954 (0.003)	0.911 (0.006)	0.877 (0.009)	0.928 (0.006)	0.827 (0.013)	0.776 (0.018)
	RPP	0.937 (0.006)	0.901 (0.012)	0.867 (0.025)	0.899 (0.014)	0.821 (0.023)	0.772 (0.009)
	Ours	0.947 (0.006)	0.916 (0.005)	0.891 (0.006)	0.923 (0.007)	0.848 (0.007)	0.795 (0.011)
D_4	None	0.895 (0.005)	0.439 (0.010)	0.396 (0.009)	0.473 (0.010)	0.431 (0.010)	0.394 (0.008)
	Restriction	0.953 (0.004)	0.912 (0.007)	0.887 (0.003)	0.930 (0.007)	0.827 (0.007)	0.773 (0.009)
	RPP	0.934 (0.007)	0.888 (0.014)	0.867 (0.014)	0.895 (0.007)	0.821 (0.020)	0.775 (0.013)
	Ours	0.949 (0.005)	**0.922** (0.007)	0.885 (0.012)	0.921 (0.008)	0.848 (0.011)	0.801 (0.009)
SO(2)	None	0.936 (0.005)	0.485 (0.010)	0.474 (0.016)	0.890 (0.006)	0.430 (0.010)	0.403 (0.021)
	Restriction	0.949 (0.002)	0.911 (0.010)	0.893 (0.009)	0.928 (0.003)	0.841 (0.012)	0.796 (0.011)
	RPP	0.935 (0.008)	0.890 (0.005)	0.870 (0.011)	0.899 (0.009)	0.821 (0.022)	0.779 (0.021)
	Ours	0.953 (0.004)	**0.922** (0.005)	**0.901** (0.005)	**0.932** (0.005)	**0.863** (0.009)	**0.823** (0.005)
O(2)	None	0.881 (0.005)	0.415 (0.008)	0.391 (0.012)	0.461 (0.012)	0.424 (0.009)	0.399 (0.014)
	Restriction	0.953 (0.005)	0.914 (0.005)	0.894 (0.005)	0.928 (0.005)	0.845 (0.011)	0.799 (0.008)
	RPP	0.931 (0.005)	0.891 (0.003)	0.861 (0.013)	0.891 (0.004)	0.824 (0.009)	0.772 (0.019)
	Ours	0.958 (0.003)	0.919 (0.006)	0.894 (0.004)	0.927 (0.004)	0.859 (0.011)	0.819 (0.010)

Maurice Weiler* and Gabriele Cesa*. General E(2)-Equivariant Steerable CNNs, *Neural Information Processing Systems (NeurIPS), 2019*
Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems (NeurIPS) 2021.*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Evaluation on MedMNIST3D dataset

- 3D voxel data

Network Group	Partial Equivariance	Nodule	Synapse	Organ
CNN	N/A	**<u>0.873</u>** (0.005)	<u>0.716</u> (0.008)	<u>0.920</u> (0.003)
$SO(3)$	None	**<u>0.873</u>** (0.002)	0.738 (0.009)	0.607 (0.006)
	RPP	0.801 (0.003)	0.695 (0.037)	<u>0.936</u> (0.002)
	Ours	0.871 (0.001)	**<u>0.770</u>** (0.030)	0.902 (0.006)
$O(3)$	None	0.868 (0.009)	0.743 (0.004)	0.592 (0.008)
	RPP	0.810 (0.013)	0.722 (0.023)	**<u>0.940</u>** (0.006)
	Ours	**<u>0.873</u>** (0.008)	<u>0.769</u> (0.005)	0.905 (0.004)

Maurice Weiler* and Gabriele Cesa*. General E(2)-Equivariant Steerable CNNs, *Neural Information Processing Systems (NeurIPS), 2019*
Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems (NeurIPS) 2021.*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Evaluation on Smoke and JetFlow simulations

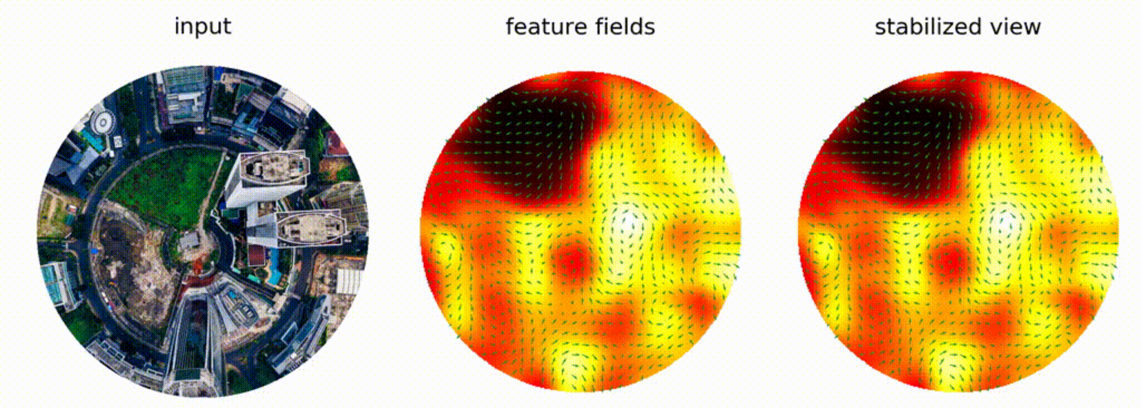
- 2D frames from simulation

Model	Smoke Future	Smoke Domain	Smoke Params (M)	JetFlow Future	JetFlow Domain	JetFlow Params (k)
MLP	1.38 (0.06)	1.34 (0.03)	8.33*	-	-	510*
CNN	1.21 (0.01)	1.10 (0.05)	0.25*	-	-	10*
e2cnn	1.05 (0.06)	0.76 (0.02)	0.62*	0.21 (0.02)	0.27 (0.03)	21*
RPP	0.96 (0.10)	0.82 (0.01)	4.36*	0.16* (0.01)	0.19* (0.01)	145*
Combo	1.07 (0.00)	0.82 (0.02)	0.53*	-	-	19*
CLCNN	0.96 (0.05)	0.84 (0.10)	-	-	-	-
Lift	0.82 (0.01)	0.73 (0.02)	3.32*	0.18 (0.02)	0.21 (0.04)	479*
RGroup	0.82 (0.01)	0.73 (0.02)	1.88*	-	-	63*
RSteer	0.80 (0.00)	0.67 (0.01)	5.60*	0.17 (0.01)	**0.16** (0.01)	185*
Ours	**0.77** (0.01)	**0.57** (0.00)	3.12	**0.15** (0.00)	0.17 (0.01)	105

Maurice Weiler* and Gabriele Cesa*. General E(2)-Equivariant Steerable CNNs, *Neural Information Processing Systems (NeurIPS), 2019*
Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems (NeurIPS) 2021.*
Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics. International Conference on Machine Learning (ICML), 2022
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Conclusion

- Complete theoretical description of the space of G-steerable filters
  - For any compact G and any transformation laws ρ_{in}, ρ_{out}
- Algorithm to explicitly construct the steerable convolution layers github.com/QUVA-Lab/escnn



- Effective way to relax hard inductive bias / learn it
  - Symmetries vary between features and scales.
  - Overconstraining leads to performance reductions.
  - CNN layers can be fine-tuned with group restrictions.

Maurice Weiler* and Gabriele Cesa*. General E(2)-Equivariant Steerable CNNs, *Neural Information Processing Systems (NeurIPS), 2019*
Gabriele Cesa, Leon Lang, Maurice Weiler, A Program to build E(n)-Equivariant Steerable CNNs, *International Conference on Representation Learning, 2022*
Maksim Zhdanov, Nico Hoffmann, Gabriele Cesa, Implicit Convolutional Kernels for Steerable CNNs. *Neural Information Processing Systems, 2023*
Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Thank you

Follow us on:

For more information, visit us at qualcomm.com & qualcomm.com/blog

Side to Side comparison: Simple MLP Setting

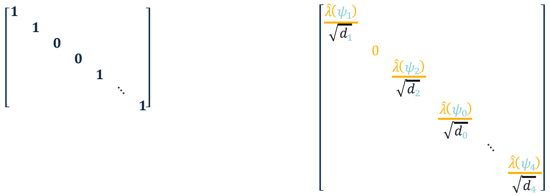
$\psi_k(\theta) = e^{ik\theta}$

- Consider MLP or 1x1 conv with $G = SO(2)$

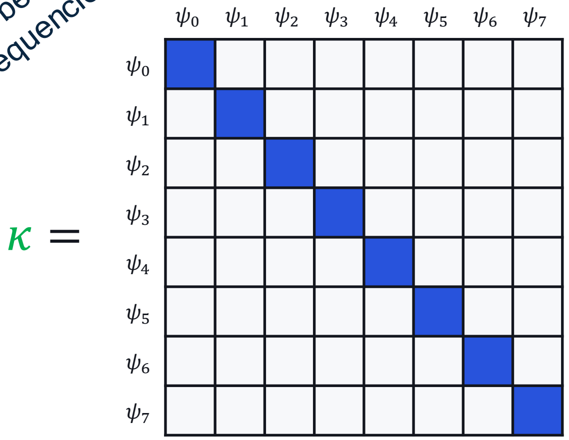
$$\kappa = unvec\left[Q_{lJ}^T \cdot Q_{lJ}\, vec(W)\right]$$

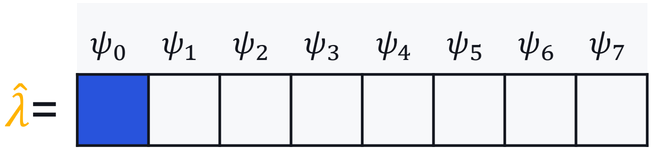
P_{lJ}

$P_{\lambda lJ}$

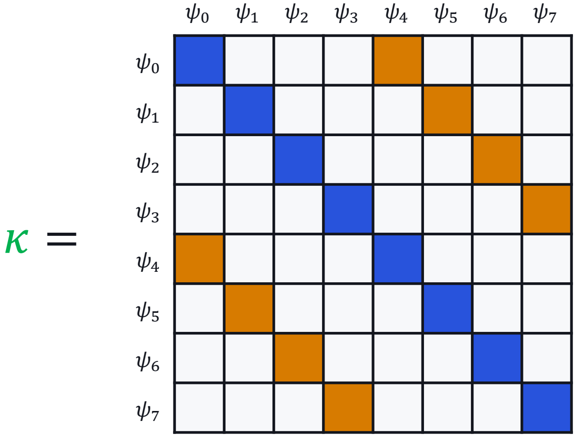


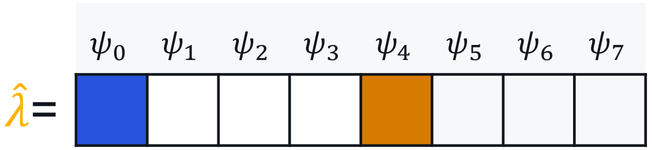
Classical Convolution Theorem: only maps between same frequencies





Breaks C_8 equivariance
Preserves C_4 equivariance
2x parameters
think of aliasing!



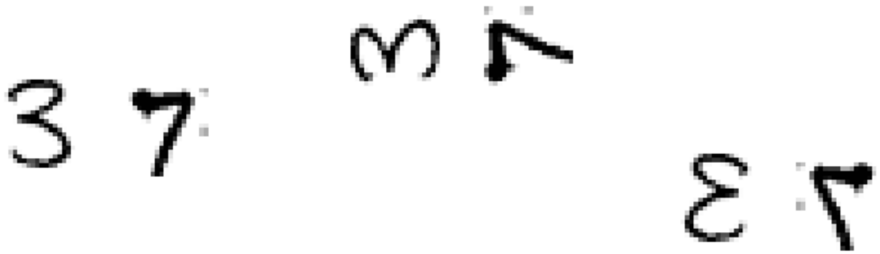


Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

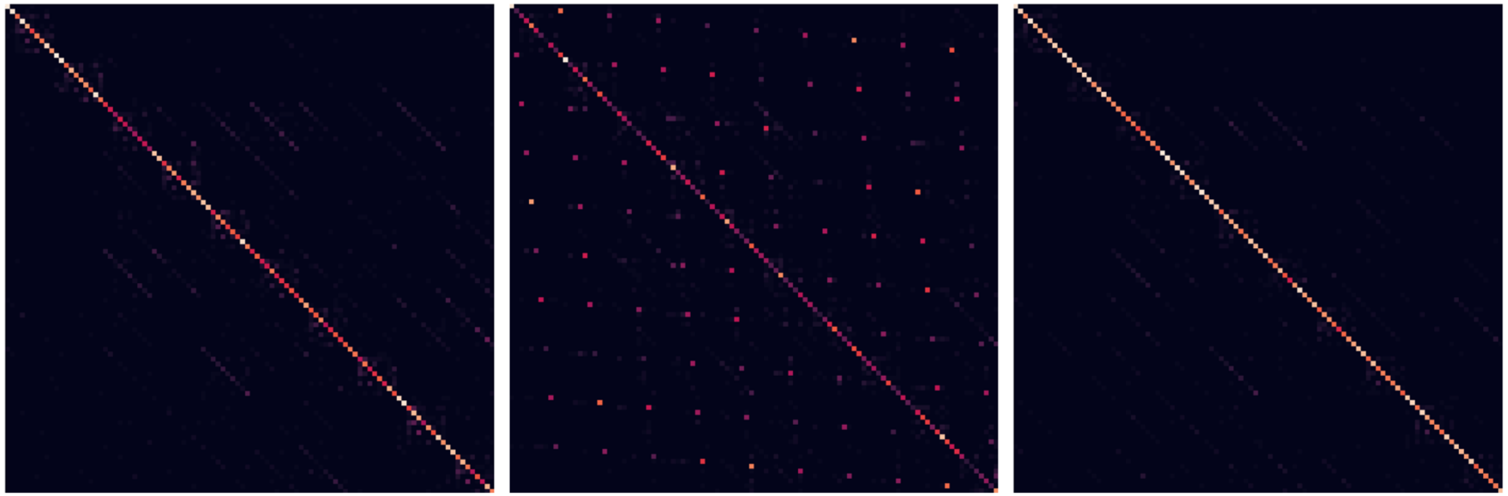
Agenda

Other experiments

Evaluation on Double-Digits MNIST



- Rectangular images containing 2 digits, independently transformed



(a) CNN (b) O(2) SCNN (c) Our O(2) PSCNN

Figure 2. Confusion matrices for `DDMNIST` with $O(2)$ symmetries. Labelled 0-99 from top to bottom and left to right.

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Effect of Band-Limiting on Double-MNIST

- Rectangular images containing 2 digits, independently transformed

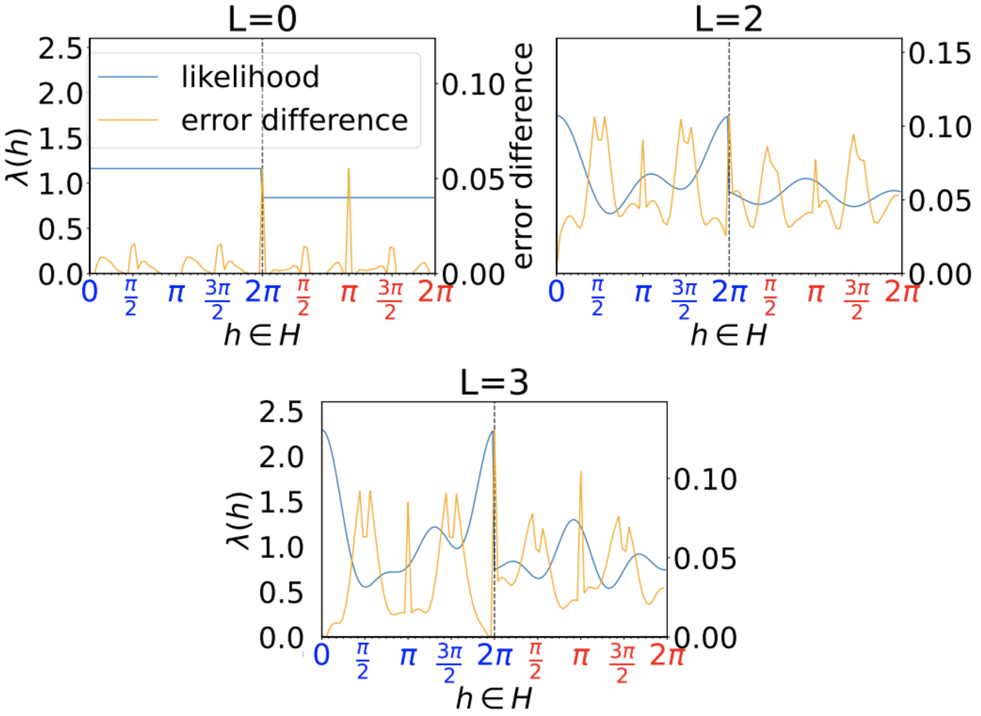


Figure 4. Likelihoods and errors of the fifth $O(2)$ PSCNN layer trained on $SO(2)$ DDMNIST under various bandlimits L.

Network Group	L	Symmetries $SO(2)$	Symmetries $O(2)$
$SO(2)$	None	0.474 (0.016)	0.403 (0.021)
	1	0.883 (0.007)	0.794 (0.011)
	2	0.901 (0.005)	**0.823** (0.005)
	3	**0.908** (0.006)	0.821 (0.002)
	4	0.904 (0.004)	0.820 (0.013)
$O(2)$	None	0.391 (0.012)	0.399 (0.014)
	0	0.469 (0.010)	0.402 (0.003)
	1	0.894 (0.011)	0.780 (0.009)
	2	0.894 (0.004)	0.819 (0.010)
	3	0.889 (0.013)	0.817 (0.007)
	4	0.891 (0.006)	0.819 (0.018)

Table 21. Double MNIST test accuracies using various levels of bandlimiting for our $SO(2)$ and $O(2)$ P-SCNNs. For each symmetry, the highest accuracy is **bold**, and the highest for each network group within this type of symmetry is underlined. Standard deviations over 5 runs are denoted in parentheses.

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Effect of Bandlimiting on Smoke dataset

Model	L	RMSE Future	RMSE Domain	Params (M) using default 3×3 kernels	Hypothetical params (M) using 5×5 kernels
RPP	-	0.81 (0.01)	0.70 (0.04)	6.43	17.24
RSteer	-	**0.78** (0.01)	**0.58** (0.00)	5.57	27.75
Ours	1	0.85 (0.01)	0.63 (0.00)	2.80	5.05
	2	**0.78** (0.01)	**0.58** (0.02)	4.30	7.84
	4	0.79 (0.02)	0.61 (0.03)	5.95	10.83
	6	**0.78** (0.01)	0.59 (0.03)	6.36	11.65

Table 23. `Smoke` RMSE scores and parameter counts (in millions, M) comparing $SO(2)$ equivariant RSteer with our $SO(2)$-PSCNN using various levels of band-limiting. Note that a band-limit of $L = 6$ equates to performing no band-limiting at all. Standard deviations over 5 runs are denoted in parentheses.

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Experiments with more effective architectures

Network Group	Partial Equivariance	L	OrganMNIST3D FourierELU	OrganMNIST3D Gated	Double MNIST with $O(2)$ Symmetries FourierELU	Double MNIST with $O(2)$ Symmetries Gated
CNN	N/A	N/A	<u>0.921</u> (0.003)		<u>0.649</u> (0.019)	
$SO(n)$	None	N/A	0.879 (0.007)	0.607 (0.006)	0.842 (0.007)	0.403 (0.021)
	RPP	N/A	0.930 (0.011)	<u>0.936</u> (0.002)	0.617 (0.043)	0.779 (0.021)
	Ours	2	0.935 (0.003)	0.902 (0.006)	0.852 (0.009)	<u>**0.823**</u> (0.005)
		3	0.932 (0.003)	0.902 (0.002)	0.853 (0.016)	0.821 (0.002)
		4	<u>**0.941**</u> (0.007)	0.896 (0.003)	<u>0.855</u> (0.004)	0.820 (0.013)
$O(n)$	None	N/A	0.821 (0.005)	0.592 (0.008)	0.860 (0.005)	0.399 (0.014)
	RPP	N/A	<u>0.936</u> (0.004)	<u>**0.940**</u> (0.006)	0.677 (0.037)	0.772 (0.019)
	Ours	2	0.911 (0.007)	0.905 (0.004)	0.869 (0.005)	<u>0.819</u> (0.010)
		3	0.920 (0.008)	-	<u>**0.885**</u> (0.003)	0.817 (0.007)
		4	0.911 (0.003)	-	0.876 (0.006)	<u>0.819</u> (0.018)

Table 24. Test accuracies on `OrganMNIST3D` and `DoubleMNIST` comparing the performance of our baseline configurations (Gated) with the structurally non-invariant configurations using a Fourier based non-linearity. For each column, **bold** indicates the highest accuracy and <u>underline</u> denotes the highest accuracy for the given network group. Standard deviations over 5 runs are denoted in parentheses.

Lars Veefkind and Gabriele Cesa. A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs. *International Conference on Machine Learning. 2024*

Agenda

Groups

Group Conv

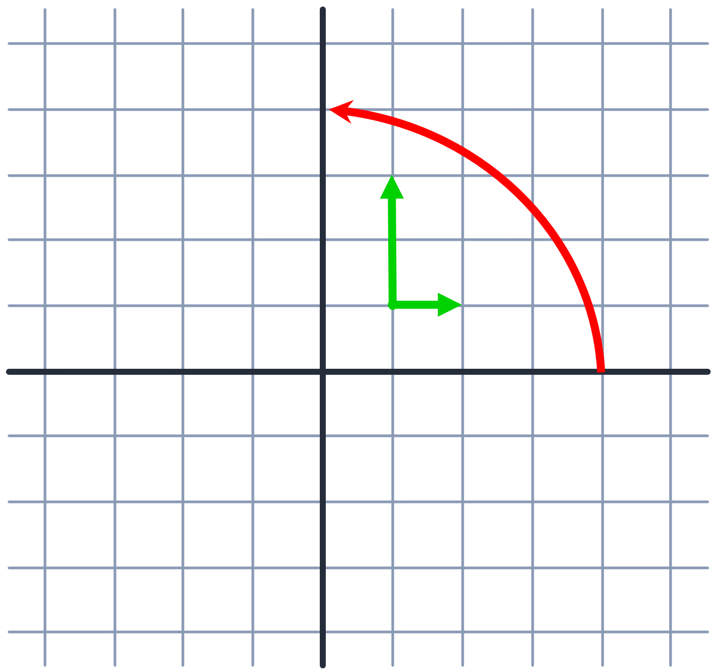
Non-Linearities

Running examples

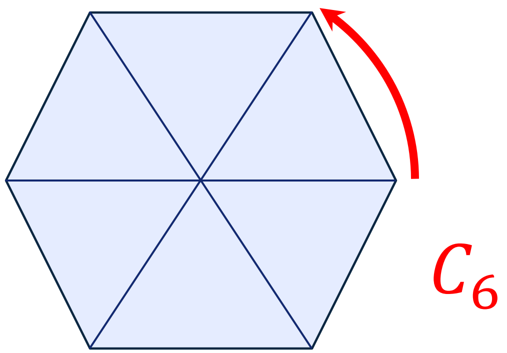
Discrete planar translations: $(\mathbb{Z}^2, +)$

Discrete planar rotations: C_n

Symmetries of squared grid: $\text{p4} = (\mathbb{Z}^2, +) \rtimes C_4$



$\text{p4} = (\mathbb{Z}^2, +) \rtimes C_4$

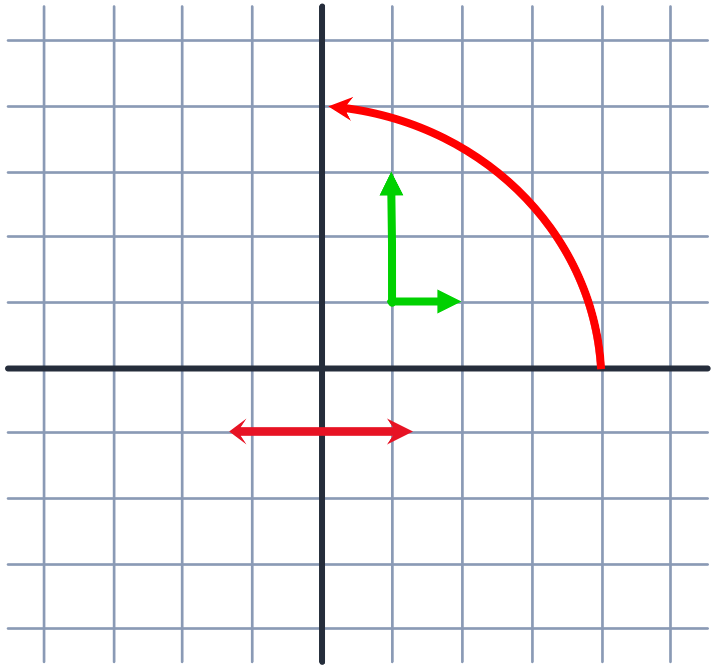


Running examples

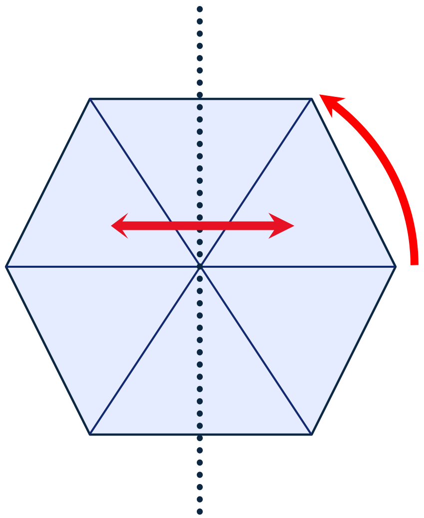
Discrete planar translations: $(\mathbb{Z}^2, +)$

Discrete planar rotations *and mirroring* : D_n

Symmetries of squared grid: $\text{p4m} = (\mathbb{Z}^2, +) \rtimes D_4$



$$\text{p4m} = (\mathbb{Z}^2, +) \rtimes D_4$$



D_6

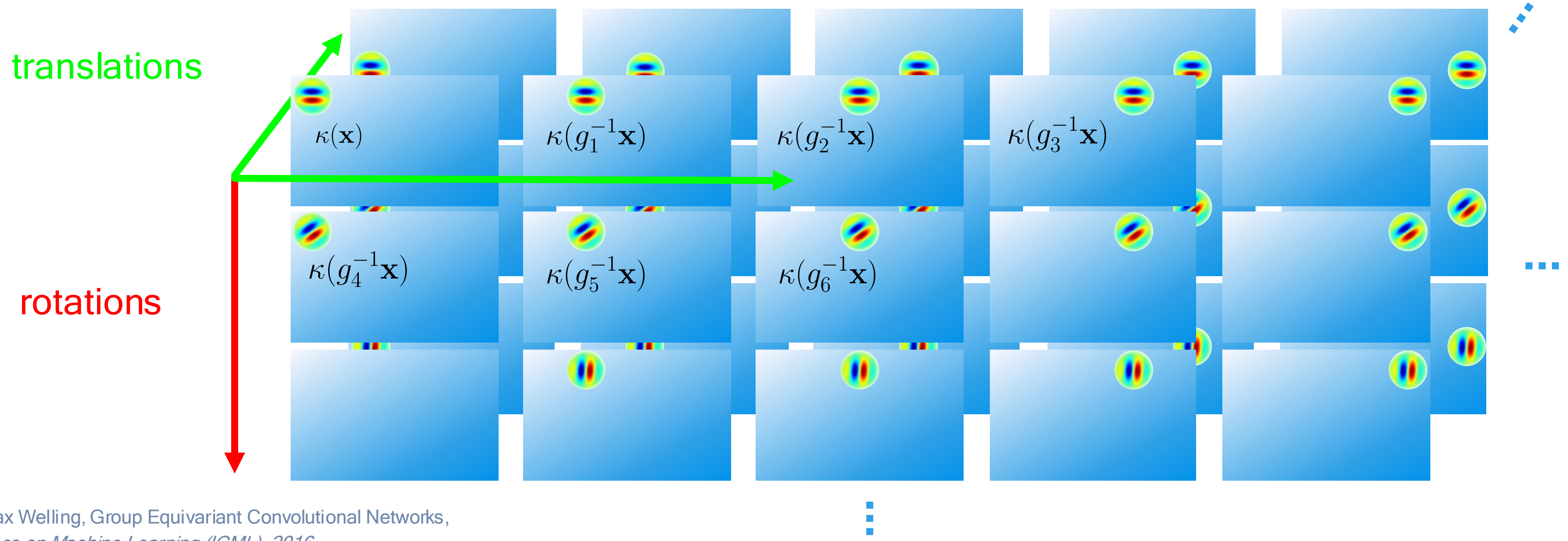
Generalize Convolution

g element of a **Group** G of transformations
(rotations, reflections, translations, …)

Group cross-correlation:

$[\kappa \star f]: G \to \mathbb{R}^{C_{out}}$ $\quad \kappa: X \to \mathbb{R}^{C_{out} \times C_{in}}$ $\quad f: X \to \mathbb{R}^{C_{in}}$

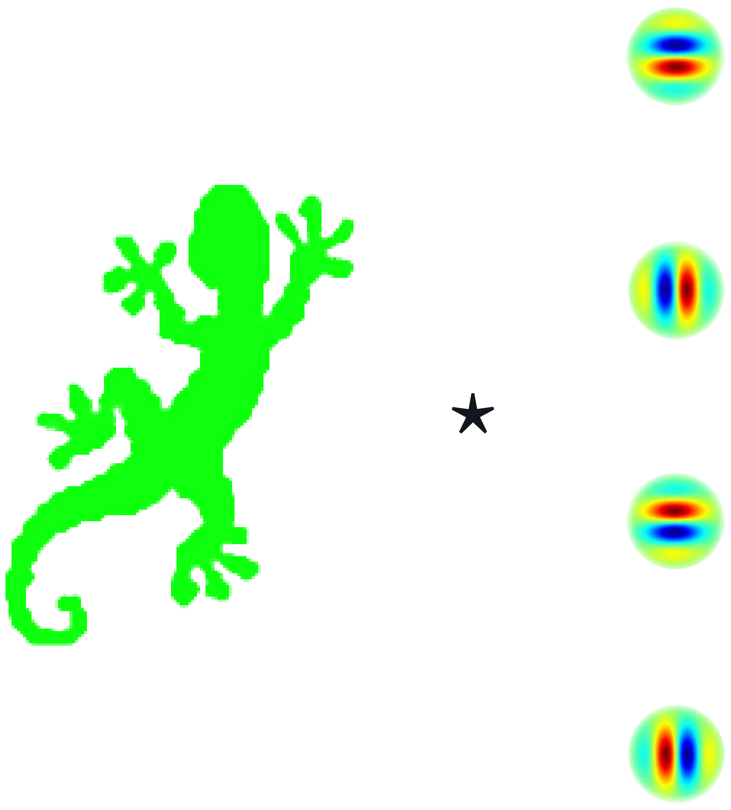
$$[\kappa \star f](g) = \sum_{\mathbf{x} \in X} \kappa(g^{-1}\mathbf{x})\ f(\mathbf{x})$$



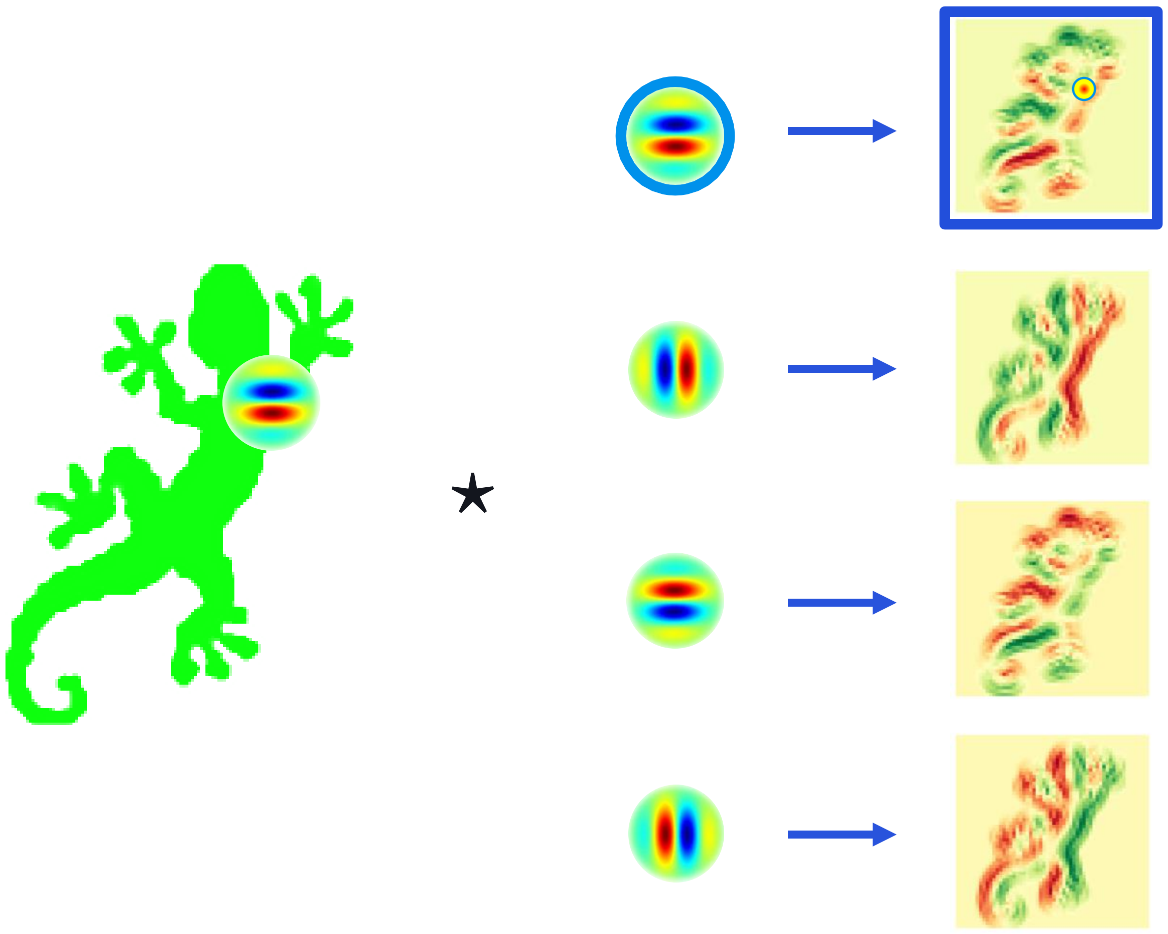
$$X = \mathbb{Z}^2$$
$$G = p4 = (\mathbb{Z}^2, +) \rtimes C_4$$

Taco S. Cohen and Max Welling, Group Equivariant Convolutional Networks,
International Conference on Machine Learning (ICML), 2016

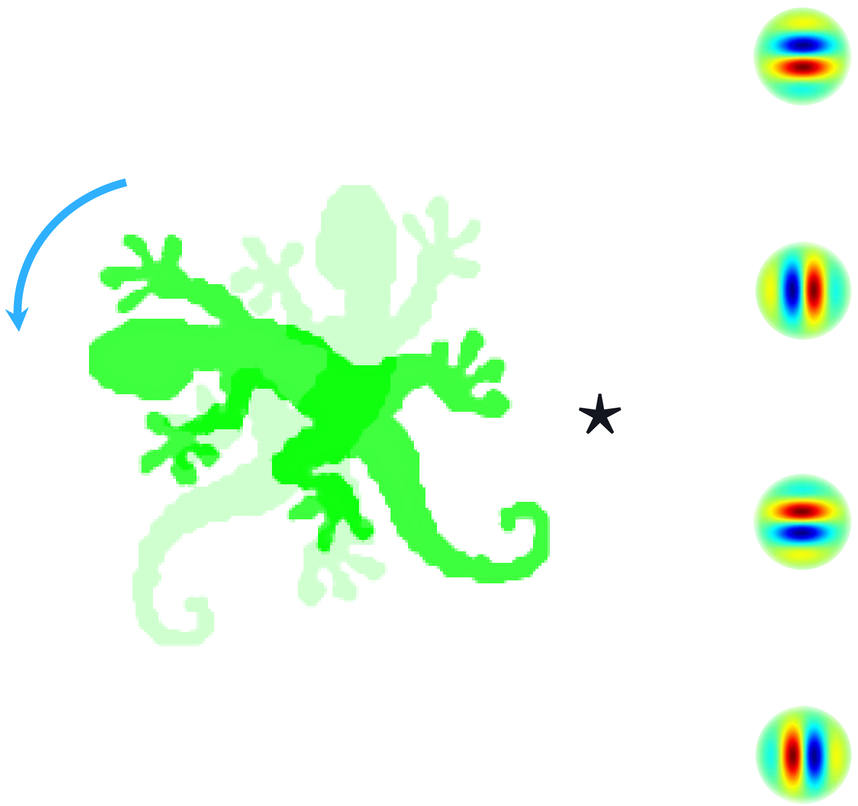
Group Convolution



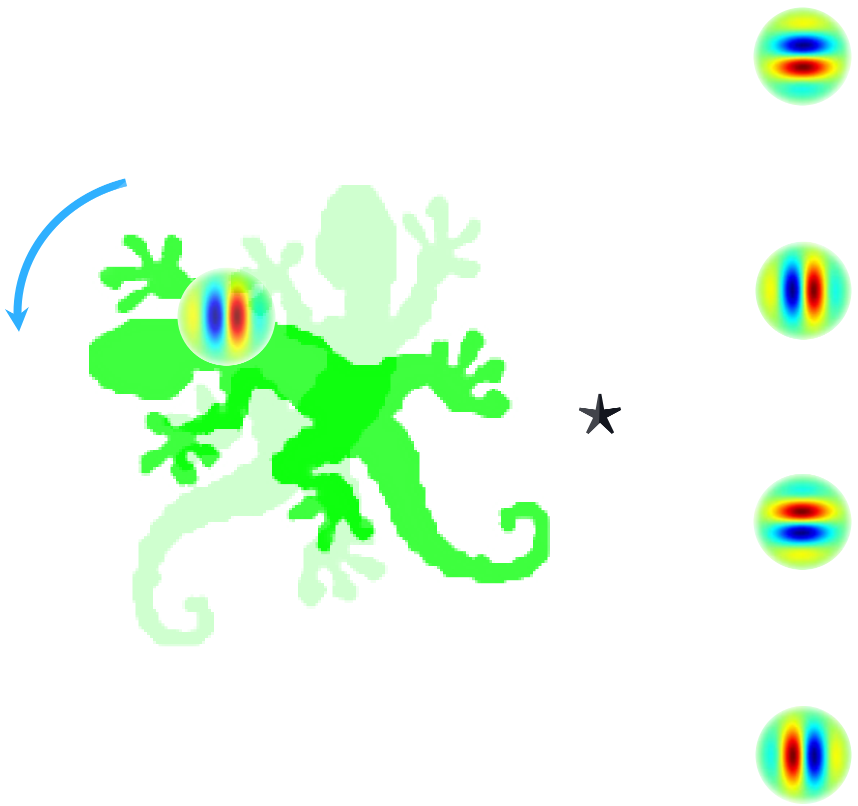
Group Convolution



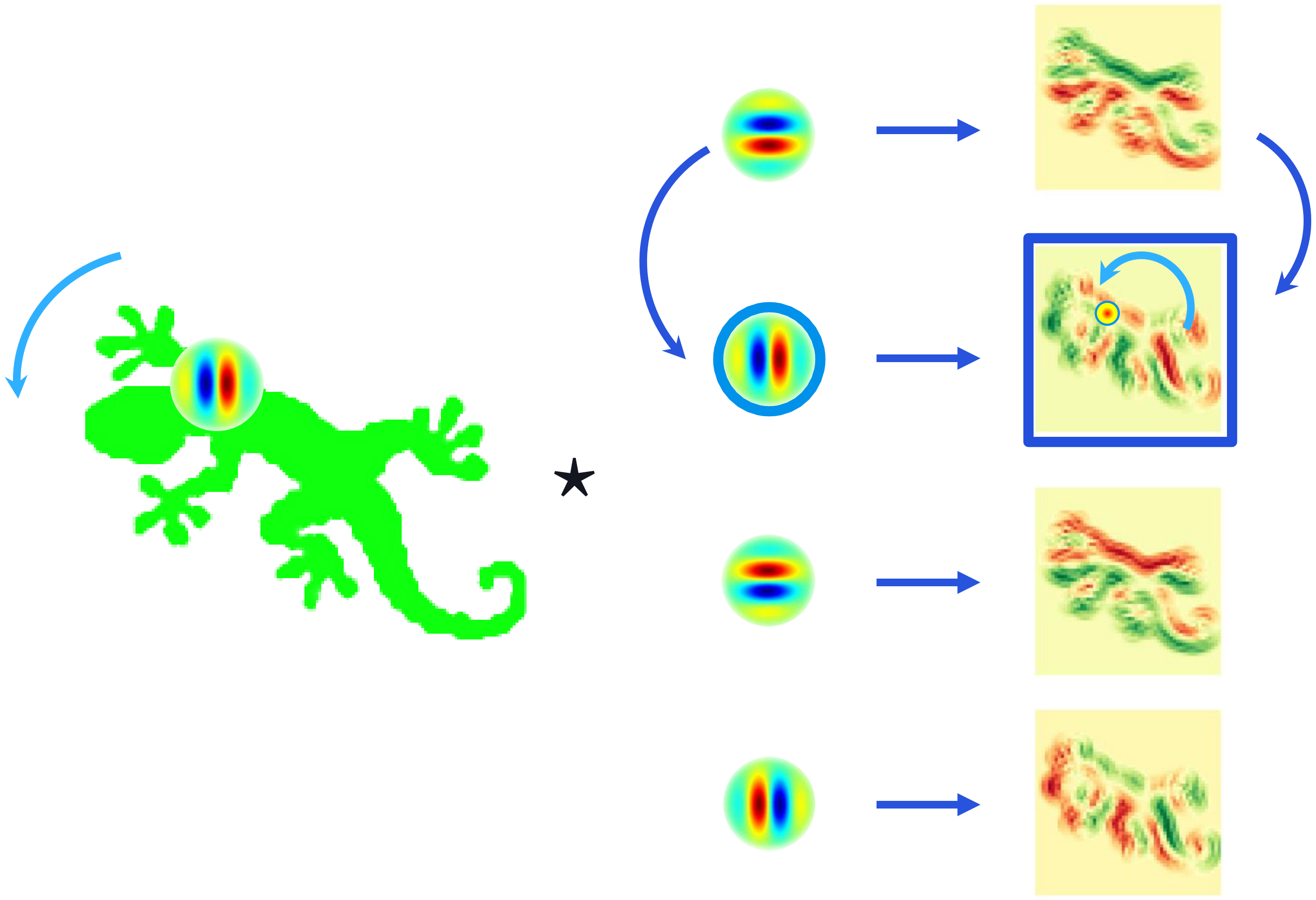
Group Convolution



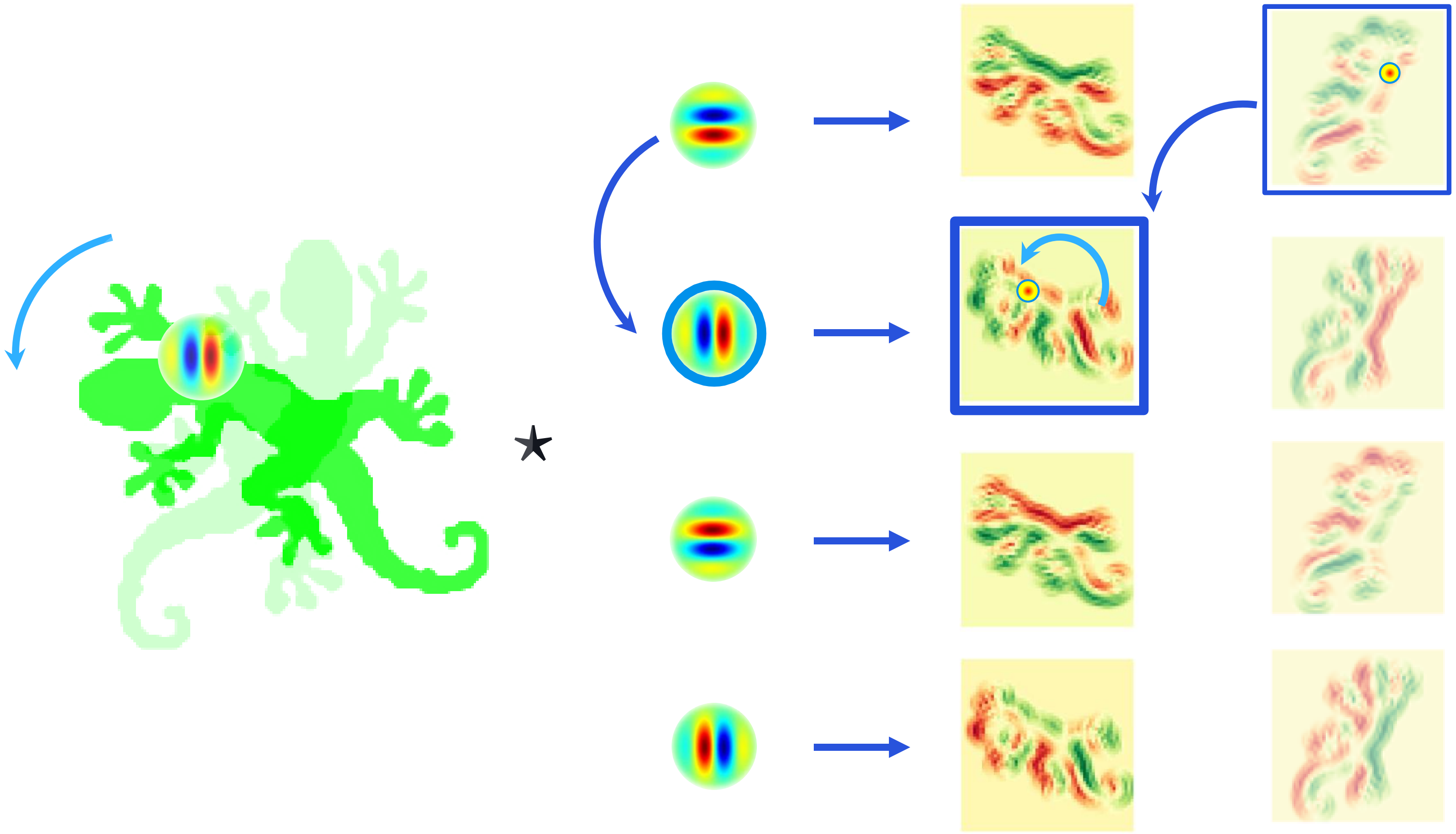
Group Convolution



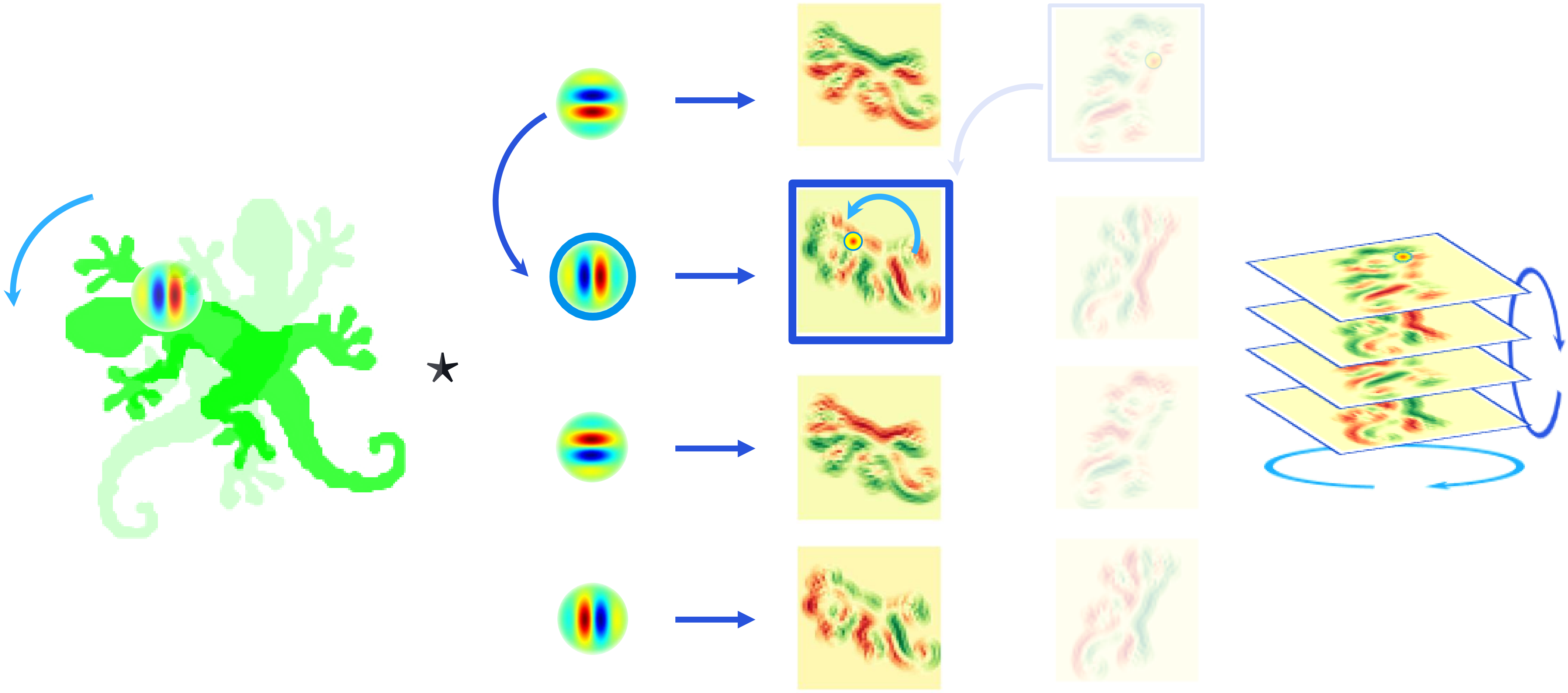
Group Convolution



Group Convolution



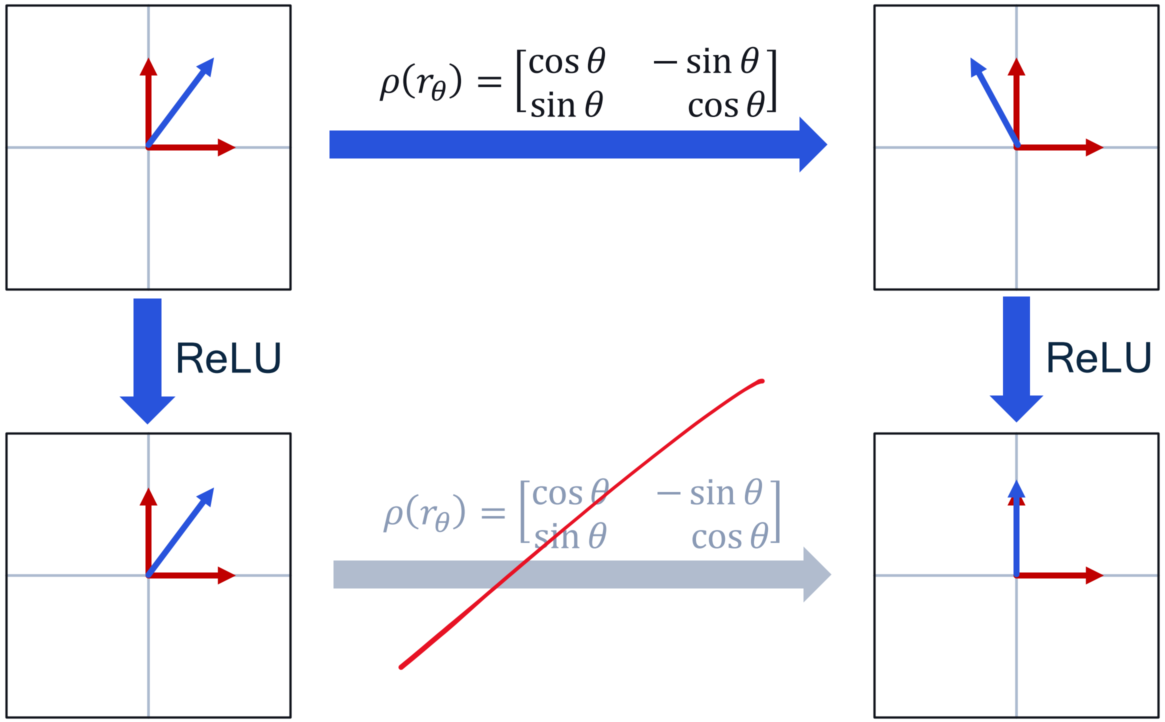
Group Convolution



Equivariant Non-Linearities

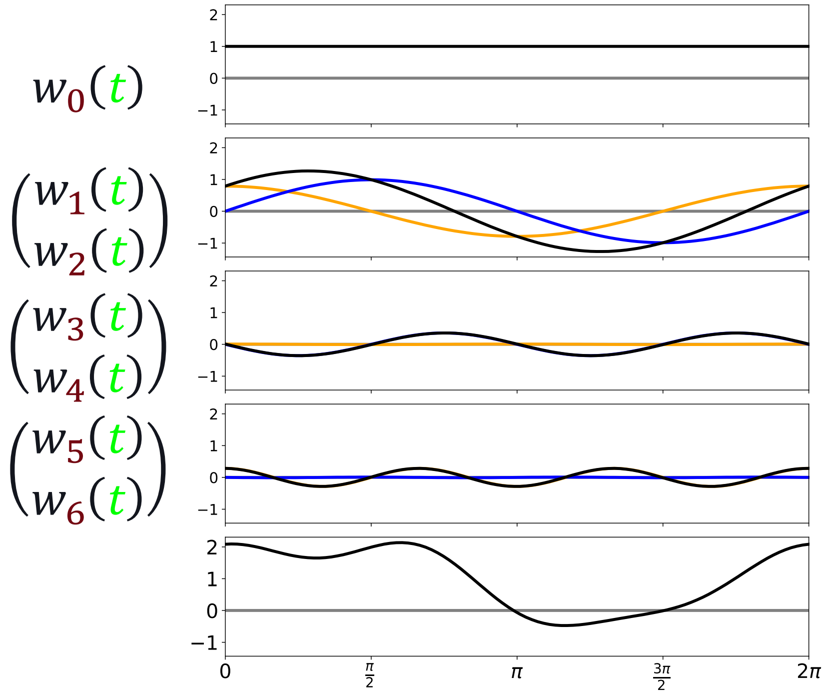
- Intermediate feature $f: \mathbb{R}^n \to \mathbb{R}^d$
- Transforms under representation of G $\quad \rho: G \to \mathbb{R}^{d \times d}$
- We can NOT always use point-wise non-linearities (e.g ReLU)

$$[g.f](x) = \rho(g)f(g^{-1}x)$$



Equivariant Non-Linearities: Fourier Transform based

- Imitate a GCNN
- Choose a band-limited subset of irreps $\{\rho_i\}_{i \in \mathrm{I}} \subset \hat{G}$
- A feature vector $f(x) \in \mathbb{R}^d$ represents a bandlimited signal in $L^2(G)$
- Apply point-wise non-linearity σ (e.g. ReLU) by:
  - Sampling the signal $f(x)$ on a finite subset $\mathcal{G} \subset G$
  - Applying σ on each sample
  - Reconstruct a band-limited signal from the samples



Equivariant Non-Linearities: Fourier Transform based

- Imitate a GCNN
- Choose a **band-limited** subset of irreps $\{\rho_i\}_{i\in\mathrm{I}} \subset \hat{G}$
- A feature vector $f(x) \in \mathbb{R}^d$ represents a bandlimited signal in $L^2(G)$
- Apply point-wise non-linearity σ (e.g. ReLU) by:
  - Sampling the signal $f(x)$ on a finite subset $\mathcal{G} \subset G$ (discrete Inverse Fourier Transform)
  - Applying σ on each sample
  - Reconstruct a band-limited signal from the samples (discrete Fourier Transform)

- Band-limit + sufficient samples to control *reconstruction error*

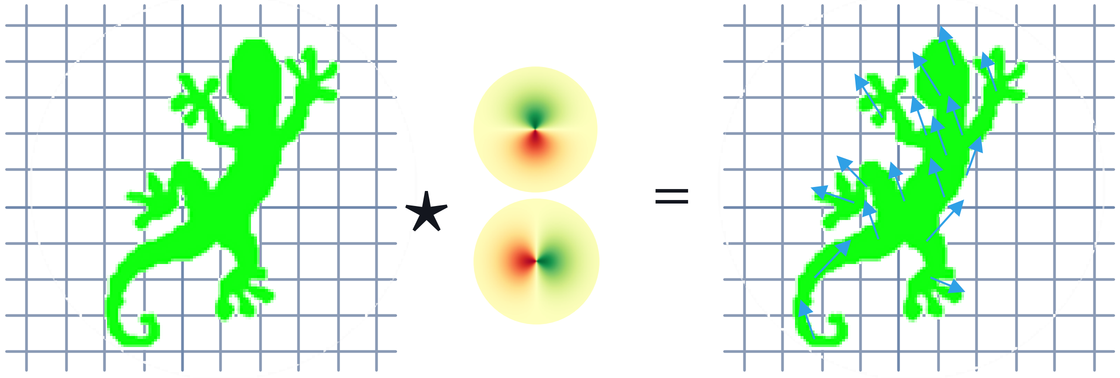
Equivariant Non-Linearities: Fourier Transform based

- Imitate a GCNN
- Choose a band-limited subset of irreps $\{\rho_i\}_{i \in \mathrm{I}} \subset \hat{G}$
- A feature vector $f(x) \in \mathbb{R}^d$ represents a bandlimited signal in $L^2(G)$
- Apply point-wise non-linearity σ (e.g. ReLU) by:
  - Sampling the signal $f(x)$ on a finite subset $\mathcal{G} \subset G$ (discrete Inverse Fourier Transform)
  - Applying σ on each sample
  - Reconstruct a band-limited signal from the samples (discrete Fourier Transform)

- Can also consider functions on homogeneous space X rather than G for reduced complexity.

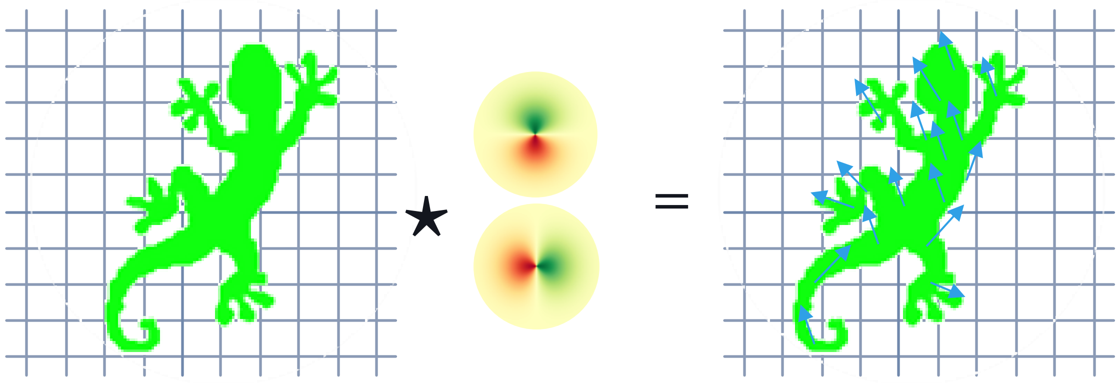
  Recall Spherical CNNs

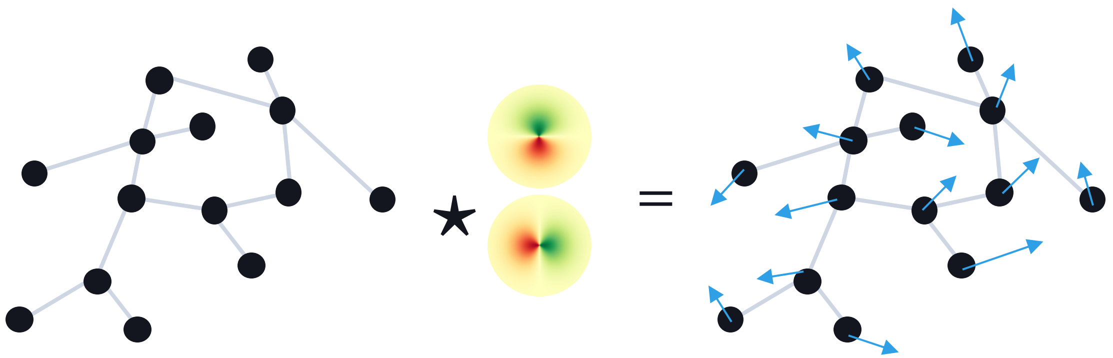
Convolution and Message Passing



$$[\kappa \star f](y) = \sum_{x \in \mathbb{Z}^n} \kappa(x - y)\ f(x)$$

Convolution and Message Passing





$$[\kappa \star f](y) = \sum_{x \in \mathbb{Z}^n} \kappa(x - y)\ f(x)$$

$$[\kappa \star f](i) = \sum_{j \in N_i} \kappa(x_i - x_j)\ f_j$$

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. (2018)